INTRODUCCION

En la actualidad, se está generando un cambio fundamental en la naturaleza y aplicación de la tecnología de los negocios. Este cambio tiene implicaciones profundas y de gran alcance para nosotros y la organización. En consecuencia, los desarrolladores en la tecnología a menudo aparecen como una barrera de acontecimientos que se presentan al azar y sin ninguna relación entre sí. Además, la mayor parte de las empresas tienen serias dificultades cuando tratan de apoyar el cambio, manteniéndose restringidas por los enfoques tradicionales para explotar la tecnología y por el legado de las inversiones tecnológicas y culturales.

Cuatro cambios importantes impactan los negocios de la actualidad:

1. El cambio en el orden económico y político del mundo es demasiado evidente para cualquiera que lee los periódicos. En realidad, nadie sabe cómo es ese cambio o dónde está sucediendo, pero éste ha venido ocurriendo desde el final de la Segunda Guerra Mundial. El mundo está en proceso de apertura y es fugaz.

2. Un cambio interrelacionado se genera en el ambiente de los negocios y en el mercado. Se acabó la competencia limitada del período estable de la posguerra. Las economías nacionales y los

mercados están transformándose. Las viejas reglas desaparecen, al igual que las barreras impuestas a la competitividad.

3. En la naturaleza de las organizaciones se genera un cambio. La corporación de antaño simplemente ya no funciona más. La transformación de los negocios, facilitada por la información, se requiere para tener éxito en el nuevo ambiente. La nueva empresa es dinámica, puede responder con rapidez a variaciones de las condiciones del mercado y tiene una estructura distinta (más plana y orientada al trabajo de equipo) que elimina la jerarquía burocrática. La nueva empresa se basa en el compromiso en vez del control. Los procesos de negocios están encausados a la productividad y a la calidad. La nueva empresa es abierta e interconectada.

4. La era de la información entra a una segunda etapa. El nuevo paradigma de la tecnología de la información va en paralelo a otros cambios. Al igual que la nueva empresa, la nueva TI es abierta e interconectada; es modular y dinámica, con base en partes intercambiables, y tecnológicamente posibilita la distribución de inteligencia y la toma de decisiones para los usuarios. Sin embargo, mediante estándares, la TI se encuentra integrada, movilizando las empresas más allá de los sistemas

aislados de la primera era. La nueva TI funciona al igual que las personas, integrando la información de datos, textos, voz e imagen en sus diversos formatos; además suministra una columna vertebral para las estructuras de negocios orientadas al trabajo de equipo; bórra las barreras existentes entre las empresas, permitiendo el restablecimiento de las relaciones externas. Más importante aún la nueva TI ha madurado hasta el punto en que es alcanzable y suministrable.

Por tanto, no solo comprender este cambio en la tecnología es llegar a una condición previa para tener éxito en los negocios en la nueva era de la información, sino emprender acciones para seleccionar los medios de hardware y software que hacen posible la realización de los objetivos de la TI. Por ello, y conscientes de la necesidad que enfrentan las organizaciones de contar con tecnología de punta, he preparado en este material la definición de parámetros específicos para la evaluación y selección de hardware y software, ya que estos elementos son indispensables para la implementación de una infraestructura de TI que proporcione una ventaja competitiva y constituya al cumplimiento de las metas organizacionales.

CONSIDERACIONES SOBRE EL NUEVO MUNDO DE LOS NEGOCIOS FACILITADO POR LA TI

2.1 El nuevo paradigma de la TI

¿Mediante cuál tecnología y cuáles medios ocurrirá el cambio hacia la nueva empresa? Es evidente que está presentándose un nuevo paradigma en la situación geopolítica del mundo. Esto genera un nuevo paradigma en el ambiente internacional de los negocios. El ascenso de la nueva empresa abierta e interconectada constituye un nuevo paradigma organizacional.

Así como las estructuras organizacionales, los ambientes de negocios y el viejo orden mundial están alterándose dramáticamente mediante progresivos cambios globales, la primera era de la tecnología de información también experimenta un destino similar. Las barreras de la tecnología están derrumbándose. Al igual que el pensamiento tradicional de la guerra fría, el viejo enfoque hacia la tecnología prueba una vez más ser inadecuado para abordar el nuevo mundo.

Una serie de estudios del grupo DMR investigó diversos cambios críticos que se presentan en al utilización de la tecnología y en al tecnología

misma. La investigación confirmó que está generándose un cambio de paradigma, está desarrollándose una nueva era tecnológica, una era que compara y se encuentra intrincadamente ligada a las evoluciones en las organizaciones y a las transformaciones más amplias en el mundo. Se entra a una segunda era de la tecnología de la información, en la cual las aplicaciones de las computadoras a los negocios, la naturaleza de la tecnología en sí misma y el liderazgo en el uso de la tecnología atraviesan por una profunda transformación. Las organizaciones que sin poder comprender la nueva era naveguen por la transición son vulnerables y se dejarán de lado.

En la actualidad, la tecnología se ha desplazado a la línea frontal en la mayor parte de las organizaciones. Se ha hecho estratégica en cuanto a que es un componente necesario para la ejecución de una estrategia de negocios.

También ha ocurrido un cambio en cuanto a quién usa las computadoras. En la primera era los usuarios principales eran los especialistas técnicos, profesionales y gerentes que diseñaban, implementaban, administraban, controlaban y a menudo eran los dueños de la infraestructura computacional de la empresa. Con la transición a la nueva era, los usuarios de negocios de la tecnología se ha colocado a la vanguardia. Los usuarios desean modelar la tecnología que se implementa en sus organizaciones, desean controlar su uso y determinar el efecto que tendrá en su propio trabajo, rápidamente comprenden que su uso efectivo de la tecnología acoplada a un cambio en la manera como ellos desempeñan sus negocios determinará su

éxito personal y organizacional, y se han convertido en la vanguardia de una revolución tecnológica de la información que con mucha rapidez altera las formas antiguas la computación organizacional.

2.2 Tres cambios decisivos en la aplicación de la tecnología de la información

En estos momentos se generan 3 cambios fundamentales en la aplicación de las computadoras en los negocios; cada uno afecta un nivel diferente de oportunidad de negocios. La tecnología de la Información hace posible que las empresas tengan una estructura de equipo de alto desempeño, para funcionar como negocios integrados independientemente de la gran autonomía de cada negocio, y alcanzar y desarrollar nuevas relaciones con organizaciones externas, con el objetivo de convertirse en una empresa ampliada.

2.2.1 Cambio 1: De la computación personal al trabajo de computación en red

Las computadoras personales (PC) han penetrado en las organizaciones para afectar casi todos los oficios. Sin embargo, su impacto rara vez puede describirse como un hecho estratégico. El principal factor limitante consiste en que el inmóvil y solitario PC por sí solo no funciona como las personas en la comunicación con los demás, en especial dentro de un grupo de

trabajo. El nuevo empuje reconoce la importancia del equipo de negocios como la piedra angular organizacional y las enormes oportunidades para apoyar los equipos dentro de la ejecución de las funciones de negocios.

El trabajo de computación en red suministra herramientas personales y da trabajo de grupo, información y capacidades para apoyar directamente todas las categorías de las personas en el sector de la información de la economía. Si los sistemas de trabajo en grupo son bien concebidos e implementados, pueden ser el punto central para el rediseño de los procesos y oficios de negocios. Esto puede generar enormes mejoramientos en la productividad y La responsabilidad. En vez de mejorar la eficiencia de una tarea como escribir un informe o preparar un presupuesto, la meta es mejorar la efectividad y el desempeño del grupo.

Los sistemas para trabajo en grupo hacen posible que los usuarios reorganicen un proceso de trabajo y cambien la naturaleza de las tareas en una unidad de negocios. En particular, los resultados representan una reducción en el tiempo de rotación para generar productos en el trabajo. La dirección también puede ahorrar tiempo, que puede reinvertirse en ejecutar actividades más importantes.

2.2.2 Cambio 2: De los sistemas aislados a los sistemas integrados

Tradicionalmente, la tecnología de la información se utilizó para ayudar a administrar y controlar los costos de las 3 fuentes: activos físicos, recursos financieros y personas. En consecuencia, sistemas aislados surgieron en 3 áreas de la organización:

- Administración y control de activos físicos e instalaciones. Estos incluían un amplio rango de sistemas de control con base en sensores o sistemas de control de tiempo real asociados con la producción y el control de procesos; sistemas involucrados con mayor almacenamiento y movimientos eficientes de materias primas o bienes en proceso o productos terminados y sistemas relacionados con el mejoramiento de la administración, la operación y la protección de las instalaciones y el equipo, que incluían plantas, ventas, vehículos, oficinas y puntos de distribución.

- Administración financiera y sistemas de control. Estos sistemas constituyen los orígenes del departamento de procesamiento de datos y abarcan la automatización de la teneduría de libros al final de las transacciones de negocios. Estaban orientados a reducir los gastos generales de negocios. Los sistemas de procesamiento de datos crecieron más allá de las aplicaciones financieras para dirigirse hacia una información más amplia, como las políticas de clientes y de seguros.

- Tecnologías para administrar y apoyar el recurso humano. Estas tecnologías tenían la intención de apoyar la administración y a otros empleados para que llevaran a cabo sus diversas funciones laborales. Incluían tecnologías administrativas, como fotocopiadoras, sistemas de automatización de oficina, como procesadores de palabras, administración de grabaciones y sistemas bibliotecarios, y comunicaciones de oficina, que incluyen teléfono, télex, correo electrónico y facsímil, y aplicaciones en los recursos humanos, como la administración de beneficios e inventarios de habilidades.

En la primera era de la tecnología de la información, las organizaciones fueron forzadas a conservar estas áreas por separado e independientes, debido a la inmadurez de la tecnología y a la incapacidad para explotarla. Esta estrategia generó la creación de tecnología aislada. Las áreas separadas de la organización (departamentos de ingeniería, función de los sistemas de información y función administrativas) asumieron la responsabilidad de estos tres diferentes tipos de sistemas de aplicaciones. Infortunadamente, a menudo los resultados fueron sistemas no integrados, muy fragmentados, que entrecruzaban sus funciones y contenidos, y cuyo mantenimiento era bastante costoso.

Debido a la madurez de los estándares de tecnología, ahora es posible planificar una arquitectura completa de la empresa. Las arquitecturas de la

empresa constituyen la columna vertebral de la nueva empresa abierta e interconectada; en efecto, éstas son un requisito clave, y permiten movilizarse más allá de la jerarquía organizacional, puesto que no son necesarios muchos de los niveles administrativos cuando la información se adquiere electrónica e instantáneamente. Tales arquitecturas hacen posible que la empresa funcione mejor como una organización coherente, suministrando información en toda la corporación para la toma de decisiones y para las nuevas aplicaciones de la empresa competitiva que trascienden las unidades autónomas de los negocios.

Al mismo tiempo, tales arquitecturas proporcionan una plataforma para la innovación empresarial en el uso de las computadoras por parte de los equipos de negocios; en tanto se mantiene la capacidad empresarial.

2.2.3 Cambio 3: De la computación interna a la computación interempresarial

En la primera era, los sistemas se consideraban como algo interno para la organización, como reflejo de las barreras existentes entre las empresas. Ahora, los sistemas se amplían a las organizaciones exteriores para vincular las empresas con sus proveedores, canales de distribución y clientes.

La tecnología se está convirtiendo en un vehículo para generar vínculos entre los socios comerciales, tanto proveedores como clientes de productos y servicios.

La nueva tecnología de alcance extendido hace posible el replanteamiento de las relaciones organizacionales externas. Los sistemas interempresariales de computación comienzan a dialogar entre sí. La cadena de valor manual de los proveedores a los clientes se convierte en una red de valor electrónico, que también vincula grupos afines (como socios comerciales) e incluso competidores. La computación corporativa se hace interempresarial, y permite el ascenso de la "interempresa".

Las tecnologías emergentes incluyen bases de datos interempresariales, sistemas de respuesta oral, mensajes electrónicos y nuevas tecnologías en el punto de venta. Estándares como el intercambio electrónico de datos (EDI) –el intercambio de documentos comerciales de computadora a computadoras entre las compañías- transforman las maneras como funcionaban las compañías entre sí.

En general, la tecnología de la información puede considerarse como clases de sistemas que van desde el nivel personal hasta los sistemas interempresariales. Las aplicaciones personales directamente apoyan y son controladas por el usuario. Las aplicaciones de trabajo se comparten en equipo o funciones que pueden localizarse de maneras centralizada o extenderse ampliamente por toda la empresa. Las aplicaciones corporativas o empresariales apoyan un amplio rango de usuarios en toda la empresa y pueden incluir muchas divisiones o departamentos. Las aplicaciones públicas o

interempresariales involucran interacción con los usuarios y con sistemas externos a la organización.

Ocho cambios tecnológicos decisivos:

1. De los sistemas de semiconductores tradicionales a los sistemas con base en microprocesadores.
2. De los sistemas con base en el anfitrión a los sistemas con base en redes.
3. De los sistemas del proveedor–propietario de software a los estándares abiertos del software.
4. Del sistema individual a los multimedia: Datos, texto, voz e imagen.
5. Del control de cuenta a la computadora proveedor – cliente con base en la libre determinación.
6. Desarrollo del software: Del arte al oficio.
7. De la interfase alfanumérica a la interfaz gráfica y multiformas del usuario.
8. De las aplicaciones aisladas a las aplicaciones integradas del software.

Crece la tecnología de la Información

La estandarización de las PC de la IBM fue el único gran factor en la creación de un crecimiento explosivo del mercado de la computadora personal y del desarrollo de una industria innovadora de software de terceras partes.

Ahora, al salir de su relativa infancia, la industria de la computación ha madurado hasta el punto de consolidarse con respecto a los estándares. Los acontecimientos que van más allá de la industria misma aceleran este reconocimiento. El ambiente de los negocios requiere arquitecturas poderosas, flexibles, con base en red, que sólo puedan lograrse mediante los estándares. La capacidad de competir y comerciar en todo el mundo está convirtiéndose en algo decisivo para el bienestar en las compañías y las economías nacionales en el ámbito internacional.

La proliferación de la computación combinada con comunicaciones instantáneas mundiales está transformando todas las industrias desde la banca hasta la manufactura. El libro flujo de información está cambiando la estructura de la economía mundial y las relaciones entre las naciones, al igual que en su interior. Existe una creciente necesidad de proveer tecnología de información en cualquier parte del mundo, y ningún proveedor puede suministrar todas las tecnologías y el apoyo asociado que requieren las empresas globales.

En respuesta a estos desarrollos mundiales, la segunda era de la tecnología de la información trae arquitecturas con base en estándares (en las cuales la tecnología se construye a partir de parte intercambiables y componentes comunes y reutilizables) provenientes de una variedad de proveedores. La computación en esta era consistirá en redes de igual a igual, inteligencia distribuida y estándares para proveedores independientes. Como la tecnología de comunicación oral, la cual permite que las personas utilicen sus

teléfonos para conectarse con otras personas en cualquier parte del mundo, la industria de la computación se desplaza para ofrecer a los usuarios conectividad igualmente transparente.

2.3 Alcanzar sistemas abiertos: Seis áreas críticas de los estándares

Aunque en sólo una década parecía remota la posibilidad de crear ambientes de sistemas abiertos, han ocurrido varios desarrollos recientes que hacen que cambie esta predicción. Las demandas del cliente, los acontecimientos políticos como el establecimiento de una nueva Comunidad Europea en 1992, la influencia de las prácticas de compra gubernamentales, el asunto de las configuraciones estándares y la previsión de proveedores líderes se han combinado para hacer el objetivo de los sistemas verdaderamente abiertos un concepto factible.

Lograr un mundo de sistemas abiertos no es un simple desafío. La portabilidad del software y la interoperabilidad de los sistemas se requieren a través de todo el espectro de la tecnología de la información, lo cual incluye ambientes operativos comunes, comunicaciones de sistemas cruzados, bases de datos compartidas y distribuidas, interfaces comunes del usuario y ambientes de desarrollo de sistemas cruzados. El DMR desarrolló estrategias para los sistemas abiertos (Strategies for Open Systems), un modelo para describir las áreas en las cuales se requieren los estándares. Este modelo se llama ambiente para los sistemas abiertos.

2.3.1 Sistemas Operativos

El sistema operativo es el nivel de software que se sitúa entre una computadora y un programa de aplicación como un sistema contable o un E-mail. Los estándares del sistema operativo son esenciales para permitir que las aplicaciones pueden transportarse a cualquier hardware y se pueda sacar ventaja de los servicios y las capacidades que éstas ofrecen.

Aunque los sistemas abiertos son mucho más que el sistema operativo Unix, éste fue importante en la creación de los estándares de sistemas operativos y, más ampliamente, en arquitecturas con base en estándares. Unix fue desarrollado a comienzos de la década de 1970 en Bell Laboratories, de la AT&T. Hacia 1990, casi todos los fabricantes lo habían adoptado como plataforma operativa para su estrategia de sistemas abiertos.

Irónicamente, la computadora personal MS-DOS incrementó el interés por Unix. El éxito del MS-DOS mostró que existían otras maneras de lograrlo en vez de hacerlo mediante una estrategia propietaria.

Unix también fue importante en el impulso para la creación de estándares formales o de jure. En particular, ANSI, del Institute of Electric and Electronic Engineers estableció un grupo de comités de estándares titulado Posix. El estándar Posix lo adaptaron todos los proveedores importantes junto con el mayor cliente de tecnología de información, el gobierno de los Estados

Unidos. Posix tiene enorme importancia en el desplazamiento hacia los sistemas abiertos.

Otros ambientes operativos con base en estándares que siguieron a Unix, en particular el OS/2, producto de la IBM, el cual es la plataforma operativa de estación de trabajo para la SAA de IBM y, como tal, desempeña un papel central e histórico, en el cambio hacia los estándares. Aunque al inicio OS/2 era propietario ya que estaba (y está) controlado por sus progenitores, se ha convertido en un estándar de facto. Además, es una plataforma operativa que estará sujeta a Posix, lo cual significa que a Posix le corresponderá hacer la especificación de lo que debe ser un sistema operativo. Esto lo conducirá a un mundo abierto o al menos cuasiabierto.

Durante la década de 1990 continuarán en observación otras importantes plataformas operativas con base en estándares de multiproveedores. Las iniciativas importantes incluyen el intento de Microsoft por reemplazar el MS-DOS y los resultados del acuerdo de la IBM con la Apple para crear una plataforma operativa "orientada al objeto".

2.3.2 Servicios de comunicación

Las grandes autopistas de información del siglo próximo ofrecerán nuevas oportunidades para que los usuarios de la tecnología de la información desarrollen y utilicen muchas aplicaciones innovadoras y creativas. Las

autopistas de comunicación abrirán mercados nuevos y existentes en áreas del mundo que antes eran inalcanzables y permitirán que las compañías utilicen mejores técnicas para alcanzar su base actual de clientes o para mejorar de manera sustancial la productividad corporativa mediante una expansión de las aplicaciones de tecnología como el EDI.

Sin embargo, estas grandes autopistas serán posibles sólo mediante estándares. Sólo cuando en realidad exista un ambiente de sistemas abiertos (que se base en estándares bien establecidos) la tecnología será capaz de ofrecer a las organizaciones una comunicación fácil y transparente, a la vuelta de la esquina o en cualquier parte del mundo.

Los estándares para interconectar varios sistemas computacionales y de comunicaciones se agrupan en 2 categorías. Puede argumentarse que los más importantes son aquéllos creados por la International Standards Organization (ISO), lo cual adoptó el modelo de 7 niveles, interconexión de sistemas abiertos (open systems interconect, OSI). Este importante y muy conocido modelo se introdujo en 1978, incluye estándares en varios niveles de comunicaciones que van desde la real transmisión física de electrones en el nivel inferior hasta los protocolos de software como el E-mail en el nivel superior. Cada nivel en el modelo utiliza servicios de los niveles inferiores y suministra servicios a los niveles superiores. La llegada de los estándares de alto nivel (llamados de aplicación) suministra algunas capacidades complejas de la segunda era.

Los estándares de comunicación posibilitarán todas las formas de comunicación. Tanto el usuario como la aplicación estarán aislados por las diferencias de varios ambientes en red. Los usuarios serán capaces de conectar computadoras de distintos proveedores en un sistema o red de procesamiento distribuido. En particular, las aplicaciones de software se basarán en estándares cliente/servidor que permitan dividir, parcelar y procesar las aplicaciones de software en la red, en vez de hacerlo en una computadora anfitrión. Las tecnologías subyacentes que posibiliten tal comunicación serán transparentes.

La conectividad transparente ocurrirá en todos los niveles en un ambiente de sistemas abiertos. Los servicios básicos de red como acceso y transferencia de archivos, correo electrónico y terminal virtual, evolucionarán hacia un conjunto muy rico de servicios ampliados que incluirá administración, directorio e intercambio de documentos de red. Los servicios de intercambio multimedia de documentos podrían incluir texto en varias fuentes, imágenes y gráficas, gráficas animadas, ecuaciones, voz, correo oral y correo oral registrado. Los formatos especializados de negocios, como pedidos de compra, facturas y cartas de crédito también se podrían hacer parte de estos servicios. Todas éstas y otras aplicaciones con base en OSI se producirán después de que se establezcan estándares específicos interindustriales para cada nivel.

Un segundo conjunto de estándares, competitivo y en algunos aspectos convergente para las comunicaciones, es el que contienen los SAA de la IBM. Estos protocolos se basan en al arquitectura propietaria de la IBM, la arquitectura de sistemas en red system network architecture (SNA), un conjunto de protocolos y servicios diseñado para las computadoras de la IBM.

Sin embargo, la dada la importancia de la IBM en la industria de la computación, SNA se convirtió en un estándar de facto, con muchas compañías que creaban ofertas compatibles con la SNA.

2.3.3 Base de datos

La manera como se almacena, maneja y se tiene acceso a la información será otra tarea importante en el establecimiento de estándares de sistemas abiertos. Los enfoques iniciales almacenaron los registros en una larga lista o jerarquía. Cuando se solicitaba información, la computadora buscaba en el archivo hasta que encontraba el registro requerido. De acuerdo con el tamaño del banco de información este procedimiento, con frecuencia, resultaba lento puesto que a la computadora le tomaba tiempo encontrar el (los) registro(s) apropiados(s). También ocasionó problemas a los programadores en la medida en que las aplicaciones de software eran independientes de la estructura de la información. Además, si por alguna razón reestructuraba la información, era necesario modificar cualquier aplicación que utilizara esa información.

El primer desarrollo importante para cambiar esta situación fue la base de datos relacional y el correspondiente lenguaje de cuarta generación. Las bases de datos relacionales tiene una estructura tabular o de matriz. Los datos se agrupaban en tablas con filas y columnas que se relacionaban entre sí. El surgimiento de los sistemas relacionales de base de datos y los lenguajes de consulta de la cuarta generación ha aumentado la necesidad de tales estándares. Un paso orientado a esta dirección es el lenguaje estructurado de consulta (structured query language, SQL). El SQL lo diseñó IBM y los proveedores y los usuarios de software lo han aceptado como estándar. El SQL ha constituido la base de datos para la compatibilidad de sistema cruzado de sistemas de base de datos para diversos ambientes.

La siguiente tendencia en la organización de datos y la construcción de sistemas se llama sistemas orientados al objeto. Estos sistemas fragmentaban la información en pedazos que contienen datos y también alguna programación que posibilita la operación de los datos. Los estándares son aún más críticos para este modelo, puesto que se necesita que los objetos sean capaces de funcionar y comunicarse con otros objetos. En consecuencia, muchos esfuerzos en los estándares están en el camino de definir estructuras y reglas para los objetos.

2.3.4 Interfaz de usuario

En la primera era, la interfaz de usuario común era un conjunto que se caracterizaba por ser brusco, críptico, difícil de aprender y de comandos artificiales que utilizaba sólo caracteres alfanuméricos. Tales interfaces de caracteres se llamaban eufemísticamente "no amigables con el usuario", aunque algunas se llamaban de manera más apropiada: "repelentes con el usuario".

Mensajes de la computadora como "sistema muerto", "entrada ilegal", "ejecutar", "matar", "abortar" y "error fatal de entrada", eran los legados de una era en la que las computadoras los utilizaban básicamente el personal de sistemas o los operadores especializados.

A medida que se hace la transmisión hacia el nuevo paradigma, cada cual comienza a apoyarse directamente en una nueva estación de trabajo de algún tipo. Aparecen en escena las interfaces de la segunda era que integran datos, texto, gráficas y voz. En 1974 apareció la primera interfaz "wysiwyg" (what you see is what you get) para procesamiento de palabras. A diferencia de sus predecesores, permitía la manipulación directa del texto y una consistente presentación del documento, en el papel o en la pantalla; la llegada de esta interfaz anunciaba una nueva era en la computación, puesto que el usuario podía centrarse más fácilmente en la tarea inmediata en vez de centrarse en las complejidades no pertinentes de operar la máquina.

En la década de 1980, una interfaz gráfica de usuario popularizada por Apple Macintosh se convirtió en una innovación histórica. La Mac permitía que el usuario operara la máquina mediante la manipulación de icono utilizando un dispositivo indicador "mouse". Cada aplicación o documento se representaba en una ventana, en la pantalla de presentación. Múltiples ventanas podían abrirse a un mismo tiempo, permitiendo que el operador trabajara en varios documentos o aplicaciones. Otras interfaces gráficas demostraron ser bastante parecidas a la Mac, aunque no hay un estándar, todas son similares. Imitando a WIMP (menús de Windows, Icons, Mouse, Pull-down) las interfaces gráficas comenzaron a suministrar una apariencia relativamente consistente para los usuarios y para desarrolladores de software.

Los estándares GUI en evolución son un requerimiento básico para los sistemas abiertos. El GUI representa la visión del usuario del ambiente de computación. Se necesita una interfaz estándar gráfica de usuario para obtener 3 beneficios claves de los sistemas abiertos: incrementar la productividad, extender el alcance de la tecnología de información a más personas, y proveer una plataforma estándar para permitir que los desarrolladores de software escriban aplicaciones compatibles y portátiles.

A medida que convergen en el mercado varios productos GUI, la "apariencia y la percepción" transforman también la tarea del desarrollador de software. La apariencia implica el aspecto de cada aplicación. La percepción se refiere a lo que ocurre cuando el usuario interactúa con la aplicación. Un GUI

estándar beneficia a los desarrolladores, quienes escriben el 80% de sus códigos para apoyar una interfaz específica de usuario. Rescribir las aplicaciones para una interfaz diferente puede ser, por consiguiente, una tarea pesada.

2.3.5 Herramientas de desarrollo de software

Ha habido muchas versiones diferentes de lenguajes populares de programación. Al poseer lenguajes estandarizados, es más fácil que los programadores desarrollen software portátil.

Los sistemas abiertos también se ampliarán a las herramientas y técnicas utilizadas para definir, crear, evaluar y mantener aplicaciones. El uso creciente de herramientas CASE, por ejemplo: destaca las ventajas de utilizar herramientas de computación para desarrollar el software. La llegada de los depósitos amplía además este apoyo. Los depósitos posibilitan que los desarrolladores de software compartan definiciones de datos y módulos de software reutilizables. También proveen un mecanismo de rastreo de principio a fin para el proceso de desarrollo de software.

El mundo de los sistemas abiertos ofrece a los programadores e ingenieros del software una variedad de herramientas de portatibilidad que permitirán desplazar o transportar las aplicaciones de un ambiente de sistemas a otro, que incluyen:

- Beneficios generales que posibilitan comparaciones de directorio, visualización del archivo de página, localización de la cadena imprimible y compresión de datos.

- Beneficios de desarrollo de software, como la creación de un archivo de referencia cruzada y tablas de símbolos para los archivos de objetos.

- Beneficios de la administración del sistema, como informes del estado de proceso de programación por lotes.

Más allá de la interfaz de usuario, otras API definen la interfaz entre la aplicación y el ambiente; por ejemplo el sistema operativo y el ambiente de comunicaciones. Las API rodean y aíslan la aplicación de los componentes del ambiente. El desarrollador presta atención a las API y a lo que hacen, y no a la mecánica subyacente del sistema operativo. El resultado radica en que la tecnología y la programación subyacentes llegan a ser transparentes.

2.3.6 Servicios de administración de sistemas

A medida que los sistemas y las redes computacionales se hacen más complejas, necesitan ser "administrados". Esto incluye la habilidad para aprender instantáneamente los problemas de la red y luego ser capaz de diagnosticar el problema y corregirlo desde la ubicación remota. Las redes

necesitan reconfigurarse y modificarse, por ejemplo, cuando se introducen nuevos grupos de usuarios. Las redes necesitan monitorearse por tiempo de respuesta y mucho más. Los modelos tradicionales de administración del sistema tienden a aplicarse a computadoras simples o, en el mejor de los casos, a vincularse a computadoras tipo anfitrión; por lo general de un solo proveedor. Con el ascenso de la computación cliente/servidor, la computadora se convierte en una red de diferentes computadoras de tamaños y marcas distintas, lo cual requiere la adopción de estándares para la administración de sistemas.

2.4 Cómo reducen los sistemas abiertos los costos de la TI

Los mayores costos del sistema de computadoras no corresponden al hardware. Este hecho con frecuencia lo omiten los diseñadores de sistemas, quienes a menudo excluyen estos componentes de sus análisis de costo-beneficio. Infortunadamente, no hay datos adicionales confiables sobre esta materia. Es probable que las más grandes inversiones se realicen en software, administración de la información, sotos humanos de entrenamiento, procesos de trabajo cambiantes y difusión general de innovación en una organización.

2.4.1 Costos reducidos en Hardware

Los ahorros pueden ser sustanciales si el hardware se convierte en un producto popular y se compra al proveedor que brinde la mejor relación

rendimiento/precio. Al carecer de control tipo cuenta, los proveedores compiten decididamente y aceptan márgenes más bajos. El cliente se beneficia.

Esto es similar a lo que ocurre en otros mercados de productos populares como el de trigo o de avena. Puesto que los productos en competencia tienen las mismas características, los compradores sólo escogen un proveedor que tenga el mejor precio para determinada media estándar del producto popular. De manera similar, cuando el software corre en el hardware de cualquier proveedor, el cliente es libre de seleccionar el hardware del proveedor que ofrezca el mejor precio para una media estándar del desempeño computacional.

De hecho, la investigación Strategies for Open Systems mostró que muchos clientes realizan "benchmarks", al evaluar diferentes equipos con los mismos programas de software para evaluar cuál se desempeña con mayor rapidez. Los sistemas abiertos también avanzan mano a mano con al tecnología del microprocesador que inherentemente posee mejor rendimiento/precio que los tradicionales semiconductores que se encuentran en los actuales mainframes y minicomputadoras.

En cierto sentido, en la actualidad existe un mercado libre para las computadoras.

La competencia alrededor de un conjunto común de estándares ha traído consigo la capacidad de centrar la investigación y el desarrollo de toda la industria en un solo conjunto de problemas. El resultado lo constituyen enormes ganancias en el rendimiento/precio del hardware.

Los ahorros en costos de hardware también son consecuencia de los sistemas escalonados. Varias partes de la organización pueden correr aplicaciones comunes aunque requieran procesadores de distintos tamaños. Con un software que funcione en máquinas con varias características de desempeño, o que pueda transportarse en ella con facilidad, el cliente puede seleccionar el sistema óptimo para determinada situación.

Algunos proveedores tienen sistemas escalonados en su línea propietaria de productos. La escalabilidad de un ambiente multiproveedor, sin embargo, ofrece mayor perspectiva en todo el rango de alternativas del procesador. La escalabilidad prevé la flexibilidad para adquirir e instalar sistemas de tamaño y poder apropiados para cada sucursal, oficina o unidad organizacional.

2.4.2 Costos reducidos de software

Una de las más grandes economías identificadas viene de la reutilización del software. Según Paul Strassmann, CIO del departamento de defensa de E.E.U.U, su mayor disminución de costo se hizo a partir de la

reutilización de componentes estándares de software y la consiguiente reducción de gastos en mantenimiento de software.

Los bloques de construcción del software se depositan como parte de una estrategia de sistemas abiertos. El depósito ya tiene más de U.S.$50 millones de componentes fabricados con los estándares DoD, que son reutilizables por medio de plataformas de hardware. Los programas piloto en la reutilización del software logran más del 60% de ahorros de codificación y mantenimiento de programas al reciclar módulos de software de asegurada calidad en un amplio rango de aplicaciones.

Una pregunta clave para el DoD es qué rapidez puede almacenar su "depósito" de software y hacer que su inventario sea fácilmente accesible. Strassmann dice que si invierten con mucha rapidez US$100 millones en desplazar el desarrollo de software de una industria de fabricación casera a un ambiente de manufactura.

Los estándares posibilitan una estrategia de software con base en la compra de paquetes de software en vez del costoso desarrollo del software individual. Considérese el mercado masivo de los paquetes de software creados bajo el estándar MS-DOS. La escalabilidad también elimina la necesidad de mantener implementaciones múltiples de la misma aplicación en arquitecturas múltiples de tecnología.

Además, debido a que las aplicaciones no están ligadas a un procesador específico, no existe la necesidad de rescribirlas cuando se usen en una máquina diferente. La portabilidad de la aplicación permite que el software pueda transportarse de una máquina pequeña a una más grande con la mínima conversión.

En la segunda era de la tecnología de información, con frecuencia el software se crea en una unidad de negocios en vez de hacerlo en una función central de sistemas. Con la portatibilidad de la aplicación, dicho software puede estar disponible para otras partes de la organización.

Por el contrario, la portatibilidad del software también puede motivar y posibilitar la coordinación e inclusive la centralización del desarrollo y mantenimiento del software, reduciendo de nuevo los costos. Un ambiente unificado de desarrollo puede producir sustanciales economías de costos.

También puede ganarse eficiencia al reducir el soporte de programación requerido por sistemas incompatibles. Las modificaciones, actualizaciones, ajustes en sistemas y aspectos similares pueden realizarse una vez y una nueva aplicación puede instalarse con facilidad en un ambiente heterogéneo de hardware.

2.4.3 Reducción de costos de administración de la Información

Los costos de administrar la información con frecuencia son una parte sustancial de la inversión general en tecnología, que incluyen los costos de diseñar bases de datos, introducir y capturar datos, establecer y mantener diccionarios de datos y administrar la información. Los cambios de la estructura de este recurso de información, que provienen de aspectos como mejoramientos en el hardware propietario, pueden plantear costos y riesgos significativos. Los sistemas abiertos tienden a reducir estos costos.

La información puede concebirse en términos de 2 componentes: el medio medio y el mensaje. Los costos asociados con el medio incluyen los costos de hardware, software y otros insumos. Los costos asociados con el mensaje pueden incluir organización, formateo, disposición, estructuración y presentación de la información de manera útil para los usuarios con problemas particulares en ambientes particulares; suma de valores en puntos de ciclos de vida de la información mediante adición, borrado y actualización del contenido de la información; y utilización o no de la información. Los costos asociados con la administración de la información (el medio y el mensaje) pueden reducirse. Muchos factores pueden afectar la magnitud de estos costos. Sin embargo, muchos de los gastos incurridos en éstas áreas se generan durante la conversión de un sistema. Debido a que las aplicaciones en un ambiente de sistemas abiertos no están ligadas a un procesador específico, no existe la necesidad de rescribirlas o de reorganizar datos cuando se convierten a una

máquina diferente. La portabilidad de la aplicación permite que el software y la información se transporten de una microcomputadora a una mainframe con un mínimo de conversión. En muchas circunstancias, el costo de transportar aplicaciones de un sistema a otro es mayor que el costo de convertir la información.

2.4.4 Costos de administración del cambio y entrenamiento del usuario

Los más descuidados e, irónicamente, los más grandes componentes de los costos del sistema son los costos humanos del cambio, que se entienden de una manera tan pobre que con frecuencia no aparecen en modo alguno en el análisis costo-beneficio en los sistemas. Descuidar estos costos es una omisión significativa para cualquier organización que intente volverse rentable en un mercado muy competitivo.

La principal función de la tecnología de información es transformar el trabajo humano y hacer que los empleados y las organizaciones sean más productivos. La TI puede transformar sustancialmente el diseño de tareas, procesos de trabajo, estructuras organizacionales, culturas corporativas y tareas individuales, Los sistemas requieren nuevos procedimientos de trabajo humano, relaciones de trabajo, flujo de trabajo, contenido de tareas, descripciones de tareas y estructura organizacional.

Los sistemas abiertos también hacen más fácil la computación para los usuarios. En un mundo de sistemas abiertos, la interfaz del usuario será cada vez más gráfica y estandarizada. El aspecto y la percepción de toda aplicación y herramienta serán similares independientemente del ambiente donde se encuentran. Las aplicaciones que funcionan de manera similar requieren menos entrenamiento y apoyo. Las aplicaciones de los sistemas abiertos también serán capaces de interoperar entre sí y proveer un amplio rango de funciones para el usuario.

Cada vez que se introduce un nuevo sistema, se requiere una cantidad de trabajo mayor tanto para los usuarios como para los implementadores. Las "conversiones" humanas entre sistemas dispares involucran el aprendizaje de una nueva interfaz del usuario, la reorganización de archivos de trabajo personal, el aprendizaje de nuevas aplicaciones y de nuevos procesos de trabajo, los cambios en el sistema significan que debe dedicarse una gran cantidad de tiempo y recursos para la administración del cambio.

La experiencia ha demostrado que los costos pueden reducirse de manera sustancial con una plataforma común, aplicaciones portables entre hardware y software heterogéneos y una interfaz de usuario común.

2.4.5 Beneficios con valor agregado a los sistemas abiertos

Además de la reducción de costos, la investigación del DMR encontró que muchas organizaciones adoptaban los sistemas abiertos para lograr beneficios con "valor agregado".

2.4.6 Reducir el riesgo de la dependencia del proveedor

Por ejemplo, hay una fuerte evidencia de que una estrategia de sistemas abiertos es menos riesgosa que una estrategia propietaria. En épocas turbulentas como ésta, un proveedor ya no satisface los criterios por los cuales se seleccionó inicialmente. Un proveedor puede quedar por fuera del negocio. Este puede ser adquirido por una gran compañía que descontinúe líneas y arquitecturas de productos. Los productos pueden salirse por la tangente y no corresponder a las demandas del cliente. Pueden perder su perfil tecnológico o ser superados por el hardware de otro proveedor. Tales cambios pueden ser en extremo arriesgados para un cliente que se encuentra limitado por una arquitectura propietaria. Estos riesgos se minimizan cuando las aplicaciones son transferibles.

La diferencia en ambientes es comparable a una situación de mercado en la que una campaña depende para sus ventas de un cliente importante, en tanto que otra tiene una variada base de clientes. El fabricante de repuestos de automóviles que depende de quienes producen automóviles, de repente puede

encontrar que sus ingresos y ganancias bajan ante la caída en las ventas de autos. Cuando más diversificada sea la compañía, será menos vulnerable.

2.4.7 Migración más fácil hacia la nueva tecnología

Es más fácil migrar fluidamente hacia una nueva tecnología con una estrategia de sistemas abiertos. Una empresa está menos ligada a su base instalada cuando aparece en el mercado una significativa innovación de hardware.

Si la base instalada es propietaria y el ascenso requiere un cambio significativo en las aplicaciones, la tendencia obvia es el renuente cambio. El peligro consiste en que la empresa puede quedarse atrás y no aprovechar las ventajas de las nuevas tecnologías. Con los sistemas abiertos, la empresa está en posición de seleccionar siempre la mejor tecnología.

2.4.8 Flexibilidad y escalabilidad de la arquitectura

Otra ventaja de los sistemas abiertos radica en que los usuarios pueden incorporar cambios organizacionales con más facilidad. Los cambios que resultan de fusiones, adquisiciones o de la expansión de las operaciones de una compañía, pueden por ejemplo, requerir que el software migre de un ambiente de mainframe a un ambiente de minicomputadora, o viceversa. Una organización que estaba muy centralizada puede adoptar una estrategia de

negocios descentralizada que involucre desplazar unidades de negocios a nuevos sitios o vincular divisiones o nuevas filiales mediante redes de área local o amplia.

A menudo, los cambios o el crecimiento organizacional pueden requerir la migración de software desarrollado para una máquina pequeña a una grande; el software que se desarrolla de una manera centralizada, o en una unidad de negocios para otra(s) parte(s) de la organización; o el software desarrollado en una organización matriz para una nueva filial. En muchos de los estudios de la organización se encontró que los sistemas abiertos suministran la flexibilidad para hacer esto más fácilmente que con los sistemas propietarios.

2.4.9 Integración del software

Los sistemas operativos, las interfaces de usuario, la red, las herramientas de base de datos y los lenguajes de programación estándares hacen más fácil integrar aplicaciones en un ambiente heterogéneo. Los mejoramientos en la integración han demostrado que generan beneficios en los negocios. En la primera era de la computación propietaria, estos cambios corporativos podrían haber sido costosos para la empresa y una pesadilla logística para el gerente de la deficiente MIS responsable de coordinar la transición.

2.4.10 Mejor selección de software de paquetes

Los proveedores independientes de software, frente a la disminución de mercados para software propietario, están desplazándose hacia los sistemas abiertos. El software para un ambiente de sistemas abiertos puede transportarse con facilidad y a menor costo a muchas plataformas de hardware diferentes. Esto aumenta ampliamente el mercado potencial para un nuevo producto de software.

Para los usuarios, esto significa que pueden escoger entre un amplio rango de paquetes de software. Las organizaciones que no sacan ventajas de este tipo de software corren el riesgo de ser superadas por los competidores.

ADQUISICION DE RECURSOS COMPUTACIONALES

3.1 Introducción

El proceso de adquisición de recursos y equipos computacionales es completos y agotador. La inversión en tiempo, dinero y esfuerzo que se realiza justifica plenamente el desarrollo de una guía de acción que oriente al responsable de conducir y desarrollar el estudio y enfatizaremos el proceso de innovación tecnológica y su aplicación a la adquisición de recursos computacionales y la definición del responsable del proyecto con otros elementos de la organización. También propone un procedimiento para lograr, con alto nivel de confianza, la determinación de los requerimientos de información, proceso que se inicia con el conocimiento de la organización y termina con un modelo para estimar los requerimientos del nuevo equipo de cómputo.

Además se sugiere un conjunto de pasos para llevar a cabo la evaluación técnica y financiera de las diferentes alternativas recibidas por parte de los proveedores, así como el establecimiento de criterios durante la decisión y negociación.

Finalmente se recomienda una serie de acciones que han llevado a cabo después de firmar el contrato con algunos proveedores en actividades tales como acondicionamiento local, capacitación y conversión y traslado de aplicaciones y datos al nuevo equipo.

Se tratará de la siguiente información:

- El proceso de innovación tecnológica de recursos computacionales.
- Determinación de requerimientos.
- Evaluación técnica de las propuestas.
- Evaluación financiera de las propuestas.
- Actividades posteriores a la firma del contrato.

3.2 El proceso de Innovación Tecnológica de Recursos Computacionales

El costo total de los departamentos de informática en las empresas está constituido por varios renglones. Una gran parte de este costo se encamina a la adquisición o desarrollo de programas de aplicación que resuelvan problemas a los usuarios. Otra parte del costo se relaciona con la operación de los sistemas computacionales y tiene que ver con las operaciones diarias, tales como captura, almacenamiento y comunicaciones de datos dentro y fuera de la organización.

Otro renglón importante está relacionado directamente con los recursos de hardware y equipo que deben adquirirse para desarrollar y operar los sistemas de información. Esta última decisión es una de las más costosas e importantes a las que se enfrenta el administrador moderno y está relacionada con la inversión en tecnología de información de naturaleza diversa.

Las decisiones relacionadas con la adquisición de recursos computacionales deberán hacerse considerando factores tecnológicos, técnicos y financieros. No obstante lo anterior, no existe procedimientos completos que quienes a los administradores de la informática durante todo el proceso de cambio de equipo, el cual se inicia con la determinación de los requerimientos de cómputo, hasta la administración de la conversión de programas y la transición y traslado de datos al nuevo sistema computacional. Con mucha frecuencia el no conocer la forma de obtener los recursos de Hardware, software y servicios de información puede resultar en un diferimiento indefinida del uso, o peor, en la adecuada selección y compra de los sistemas incorrectos.

Normalmente el director o gerente de informática de las organizaciones, apoyándose en los departamentos de soporte técnico, operaciones y/o desarrollo de sistemas de aplicaciones quien coordina el proceso de cambio de equipo computacional. Además se requiere que el usuario de las diferentes áreas que integra la empresa se involucre durante el proceso de cambio de equipo. De hecho el procedimiento que se propone a lo largo de este tema involucra al

usuario y presupone que los departamentos de apoyo participan durante la vida del proyecto.

Es importante conocer los síntomas que se observan en las operaciones, y que constituyen un detonante para el proceso de cambio de equipo, a fin de que el administrador de los recursos de hardware se prepare para el manejo del proyecto. Estas causas pueden incluir las siguientes:

- Problemas de servicios con el proveedor actual. Esto se refiere a que la empresa no está obteniendo el servicio que esperaba adquirir equipo computacional, por ejemplo: la empresa necesita que se le dé mantenimiento al equipo actual y el proveedor no cumple con la fecha indicada para ello y por eso desea cambiar de proveedor y de equipo.

- El equipo computacional actual es obsoleto. La empresa no se ha actualizado en lo que se refiere a compra de equipo de cómputo.

- Saturación y falta de capacidad del equipo computacional. El equipo con el que cuenta la empresa ya no es suficiente para hacer frente a las necesidades de tecnología de información.

- Necesidad de incorporar nuevos sistemas de aplicación a la organización. El medio ambiente está demandando un mejor uso de la tecnología de información

de lo que lleva consigo la necesidad de que la empresa utilice nuevos sistemas que requieren capacidades de equipo diferentes a las actuales y por ello es necesario adquirir nuevo equipo.

- Haber tomado una decisión equivocada durante el proceso de selección del equipo actual. Si las decisiones referentes a la adquisición de equipo computacional no han sido las adecuadas a las necesidades de la organización, será necesario adquirir nuevo equipo.

- Cuando se tienen requerimientos de competitividad que no puedan ser alcanzados con la tecnología actual. En estos casos, para estar a la par con la competencia es necesario que la empresa adquiera nuevo equipo.

- Cuando existe la necesidad de cambiar la filosofía de operación de los sistemas actuales, tales como: cambio de archivos convencionales a bases de datos, cambio de operación de batch a operación en línea, o bien cuando se desea emigrar de un ambiente centralizado a un ambiente de redes de microcomputadoras o estaciones de trabajo (downsizing) o viceversa.

3.3 Determinación de requerimientos

La primera fase consiste en determinar los requerimientos del nuevo equipo que será adquirido, a fin de poder transmitir las necesidades de manera clara a los

diferentes proveedores. Para lograr con éxito lo anterior es necesario llevar a cabo algunas actividades que se han denominado previas a la aplicación de la presente guía.

El modelo que se presenta en esta sección tendrá una utilidad singular para el administrador de la función de informática, ya que facilitará su labor de cálculo en forma considerable. Los especialistas en planeación de las capacidades de equipo de cómputo invierten una gran cantidad de tiempo y esfuerzo en crear a detalle modelos computacionales altamente complejos, para determinar la configuración que se aproxime a la mejor de las cargas de trabajo futuras de la organización.

3.4 Actividades previas a la determinación de requerimientos

3.4.1 Conocimiento de la organización:

El primer paso recio que deberá realizar el responsable del proyecto de cambio de equipo, es tener un conocimiento profundo de la organización o la entidad de negocios que recibirá el servicio que será adquirido. Por ejemplo: si el equipo va a ser adquirido por una empresa que se dedica a la manufactura de bienes de consumo, el responsable del proyecto deberá conocer las áreas de ventas, administración, mercadotecnia, producción, finanzas y recursos humanos. Si el equipo será adquirido por una empresa para dar servicio a la división de metales, por ejemplo, entonces el responsable del proyecto deberá conocer a

fondo las entidades que conforman la división metales de la empresa. En caso de que la organización decida contratar los servicios de un consultor para llevar a cabo el proyecto de cambio de equipo, habrá que tener cuidado de proporcionarle toda la información, o bien, que se involucre con cada una de las entidades funcionales.

El conocimiento de la organización implica por otra parte, conocer el plan general o estratégico del negocio. Este plan en la mayoría de las empresas es elaborado por los departamentos de finanzas y administración, o bien, por el área de planeación estratégica. Cabe recalcar que es importante para el responsable del proyecto de innovación tecnológica, conocer el índice o porcentaje de crecimiento del negocio estimado durante el horizonte de planeación del proyecto.

3.4.2 Plan de desarrollo de aplicaciones:

El siguiente paso previo a la determinación de requerimientos de equipo es contar con un plan de desarrollo de aplicaciones a corto, mediano y largo plazo. La idea de este punto consiste en conocer las aplicaciones que serán desarrolladas y que operarán en el nuevo sistema computacional durante el horizonte de planeación del proyecto. El horizonte de planeación se define como el lapso de tiempo futuro que se considera en un análisis. Si se desea, por ejemplo, un equipo o un sistema computacional tenga por lo menos un tiempo de vida cuando menos de 5 años, entonces se debe conocer las aplicaciones a que estarán operando durante los próximos 5 años, es decir, el horizonte de planeación del proyecto de

cambio será de 5 años. Normalmente el responsable del proceso de desarrollo de sistemas en la organización es el jefe de desarrollo de aplicaciones o análisis y programación, éste deberá ser el responsable de desarrollar el plan de desarrollo de aplicaciones antes de iniciar el estudio de determinación de los requerimientos del nuevo equipo.

3.4.3 Filosofía de operación o tipo de solución requerida:

El plan deberá incluir aspectos tecnológicos requeridos para el desarrollo de las nuevas aplicaciones, tales como base de datos, códigos de barras, sistemas batch o en línea, ya que estas especificaciones pueden modificar de manera sensible los requerimientos y restricciones a considerar en el nuevo equipo. La filosofía de operación que se desea con el nuevo equipo requiere un análisis del tipo de solución que se implantará con el nuevo equipo que se van a adquirir.

3.4.4 Requerimientos opcionales

Los requerimientos opcionales constituyen el conjunto de características que serán de gran utilidad si se encuentran presentes en el equipo, pero el hecho de que no se cumplan totalmente, no implica que la propuesta del proveedor será descartada. Ejemplos de estos pueden ser los siguientes:

- La existencia de usuarios con configuraciones a la que se está proponiendo y que se encuentren en localidades geográficas cercanas para tener un soporte mutuo.
- Disponibilidad de algún sistema de aplicación o paquete ya desarrollado para asegurar una implantación rápida y exitosa.
- Alto grado de satisfacción de los usuarios actuales.

3.5 Requerimientos futuros de las aplicaciones actuales (RFA)

Los requerimientos que se especificarán en el equipo nuevo, requieren de proyectar la utilización de los recursos actuales durante el horizonte de planeación del equipo. Para esto puede usarse la siguiente fórmula:

$$RFA\ i = RAA\ i * (1 + PC/100)\ HP$$

En donde:

RFA i: es el requerimiento futuro proyectado en las aplicaciones actuales del recurso i.

RAA i: es el requerimiento actual en las aplicaciones actuales del recurso i.

PC: es el porcentaje de crecimiento esperado del negocio.

HP: es el horizonte de planeación del proyecto de cambio.

Tal como se menciona en las actividades previas, debe recalcarse que el porcentaje de crecimiento del negocio debe ser proporcionado por el departamento de planeación estratégica del negocio, o el equivalente, dependiendo de la estructura organizacional de la empresa. Este valor deberá de estimarse con detenimiento, ya que constituye la columna vertebral de los cálculos efectuados durante la fase de determinación de requerimientos.

La fórmula anterior puede utilizarse para calcular el área total que ocuparán, al final del horizonte de planeación, los archivos de disco de todas las aplicaciones que se encuentran funcionando actualmente en la compañía, tomando como base el porcentaje de crecimiento del negocio.

3.6 Formato de la propuesta que se recibirá de los proveedores concursantes

Es importante definir estándares de la propuesta que se ha de recibir de los diferentes proveedores de soluciones, con el fin de facilitar la evaluación técnica y financiera de estas propuestas y, por ende, de la decisión final.

El formato de la propuesta podrá variar de una empresa a otra o dependiendo del problema específico que deba ser resuelto utilizando la tecnología de la información. No obstante lo anterior, se incluye una lista de parámetros que pueden ser incluidos en las propuestas que entreguen los proveedores.

3.6.1 Sistema o solución configurada. Este punto normalmente es presentado con un adecuado nivel de detalle por la mayoría de los proveedores. Sin embargo es importante cerciorarse que incluya la descripción técnica detallada de la solución propuesta y las capacidades de crecimiento del sistema. Esta información debe incluir manuales técnicos de hardware y software configurado, así como diagramas esquemáticos de la configuración propuesta.

3.6.2 Requerimientos de instalación. Los requerimientos de instalación de los equipos pueden cambiar una decisión determinada, ya que afectan el costo total de la solución. Se debe recordar que existen soluciones en el mercado que tienen altos costos en requerimientos de instalación, como por ejemplo, los mainframes. Esta información debe incluir requerimientos de:

- Espacio físico que ocupa el equipo.
- Instalaciones eléctricas y equipos de reguladores de voltaje.
- Temperatura ambiental y equipo de refrigeración.
- Requerimientos especiales tales como piso falso, equipo de control de humedad y en general todos los requerimientos de instalación del equipo.

3.6.3 Soporte del proveedor. Soporte del proveedor es tan importante como el producto dentro del cuadro global. Desgraciadamente con frecuencia esto no se

considera hasta después de que el equipo ha sido adquirido. La información que se requiere por parte del proveedor en este renglón incluye:

- Soporte para el entrenamiento del personal en el nuevo equipo y calendario de cursos, incluyendo su costo.
- Personal de soporte para hardware, software y en general, para el mantenimiento del equipo computacional.
- Inventario de equipos de respaldo compatibles con el equipo configurado en la propuesta.
- Apoyo y experiencia en la conversión de las aplicaciones y de los programas de aplicación que han de ser convertidos al nuevo equipo computacional.

3.6.4 Información de costos. En este punto se incluye toda la información económica y financiera de las propuestas de equipo computacional. Lo anterior comprende precios, plazos de pago y opciones de compra disponibles por parte del proveedor y, en general, todos los datos requeridos para desarrollar la evaluación económica del proyecto de inversión.

3.6.5 Condiciones del contrato. Las condiciones del contrato por lo general se especifican en formatos fijos que el proveedor anexa en la mayoría de las propuestas. Este formato comúnmente es elaborado por el departamento legal de la compañía.

3.6.6 Nivel o grado de cumplimiento de los requerimientos obligatorios y opcionales del cliente que tiene la solución presentada por el proveedor. En este punto es importante insistir a los proveedores deberá incluirse en la propuesta entregada en forma expresa y por separado.

3.7 Factores que deben evaluarse

La evaluación técnica de las propuestas debe realizarse exclusivamente sobre las soluciones que cumplen al 100% los requerimientos obligatorios y que cumplen mejor los requerimientos opcionales.

En término generales, los factores que deben considerarse durante la evaluación técnica de la propuesta se clasifican en factores de hardware, de software y de proveedores.

3.7.1 Factores de Hardware. Son las características relacionadas con los componentes físicos de la computadora y sus indicadores más representativos, tienen que ver con las capacidades y velocidades de los diferentes componentes. Estos componentes incluyen procesador, memoria RAM, discos, terminales, impresoras, etc.

3.7.2 Factores de software. Este renglón se refiere al software interno o software de sistemas, el cual se componen de programas de control, incluyendo el sistema operativo, software de comunicaciones y administrador de base de datos.

Además se consideran paquetes especiales, tales como simuladores, análisis financieros, programación lineal, control de proyectos, análisis estadísticos y paquetes que se enfocan a resolver problemas funcionales a usuarios, tales como contabilidad, cuenta por pagar, facturación, etc. Finalmente es importante destacar que en la mayoría de los casos el tiempo de vida útil del software de aplicación es superior al tiempo de vida útil de las plataformas de hardware. Esto nos lleva a la necesidad de llevar a cabo constantes conversiones y adaptaciones al mismo sistema de información para que corra funciones en plataformas diferentes y más modernas de hardware. Como consecuencia de lo anterior, se recomienda durante la definición de los factores de software otorgar un peso fuerte a la posibilidad y facilidad de que todo el software que se desarrolle sea abierto o transportable fácilmente a diferentes plataformas de hardware. Si por el contrario, el software ofrecido por un proveedor sólo permite el desarrollo de sistemas cerrados de aplicación, el cliente será dependiente de con respecto al proveedor durante muchos años debido al costo de convertir posteriormente todos los programas a otras plataformas. Un ejemplo de lo anterior puede ser un sistema de nómina cuya vida útil sea de 10 años y opere en un equipo que no durará todo ese tiempo. Este sistema debe ser lo más abierto posible para permitir el cambio de equipo sin que se afecte su funcionamiento.

3.7.3 Factores de proveedor. El tercer grupo de factores que se deben tomar en cuenta durante la evaluación técnica de la propuesta se refiere al proveedor. Incluso algunos autores consideran que el soporte es el segundo criterio de importancia para tomar una decisión de compra, sólo después de la adecuación

de la computadora para los propósitos de la empresa. Otros otorgan un peso mayor al servicio que al precio y al rendimiento.

Un aspecto importante que debe considerarse es la tendencia a que mientras más tiempo permanezca un cliente con un mismo proveedor, especialmente si no existe compras importantes, el servicio y el soporte del proveedor se decrementa.

Todo lo anterior justifica la necesidad de analizar con mas detenimiento los posibles factores a considerar.

Con relación al servicio y al proveedor. A continuación se analizan algunos de los siguientes factores:

3.7.4 Generalidades del proveedor: en un sentido amplio, las generalidades del proveedor es toda aquella información relacionada con la imagen que tiene el proveedor en el mercado local, regional y mundial, considerando aspectos técnicos, mercadológicos y financieros, de tal manera que aseguren la permanencia del proveedor. Los factores a evaluar incluyen:

- Representación mundial y regional del proveedor.
- Tiempo de entrega del equipo ofertado y futuras ampliaciones.
- Profesionalismo y preparación de los vendedores.

- Situación económica y financiera del proveedor.
- Calidad de la documentación y manuales disponibles.

3.7.5 Apoyo en la capacitación: Este aspecto a considerar en los proveedores es el apoyo que brinda en la capacitación al personal técnico y usuarios en el uso de los recursos de hardware y software propuestos. Estos factores incluyen:

1. Capacitación al personal de las áreas de investigación y soporte técnico.
2. Capacitación al área de análisis y programación.
3. Capacitación a proveedores.
4. Capacitación a usuarios.

Es importante resaltar que los costos de capacitación pueden variar de manera significativa de un proveedor a otro.

3.7.6 Mantenimiento de hardware y software: El mantenimiento es otro de los factores cruciales a tomarse en cuenta durante la evaluación técnica de las propuestas. Consiste básicamente en la capacidad de un proveedor para proporcionar un soporte y servicio adecuado, para asegurar el funcionamiento continuo del sistema computacional. Algunos de los aspectos a considerar son los siguientes:

- Calidad y cantidad de personal capacitado de tiempo completo disponible en hardware y software.
- Tiempo promedio que tarda el proveedor en atender las fallas reportadas.
- Tiempo o porcentaje de sistemas funcionando, que es el tiempo que dura trabajando el sistema sin que ocurra algún problema.
- Tiempo promedio que el sistema permanece caído durante cada falla.
- Existencia de una bodega cercana con partes y repuestos para responder con mayor oportunidad a las fallas.
- Existencia de horario extendido, siete días a la semana y 24 horas diarias para atender fallas.

3.7.7 Equipos de respaldo: los equipos de respaldo disponibles por parte del proveedor u otros usuarios constituyen un aspecto a considerar durante la evaluación de los proveedores. Estos equipos pueden soportar la operación (cuando menos de forma parcial) de la empresa del cliente durante fallas prolongadas. Obviamente el cliente puede contar con una infraestructura de sistemas de respaldo, sobre todo cuando la tendencia en el desarrollo de sistemas es en línea, lo cual hace que re quiera soporte operacional continuo.

Algunos de estos factores pueden incluir.

1. Cantidad de equipo de respaldo existentes.
2. Facilidad de trasladar procesos a los equipos de respaldos.
3. Grado de satisfacción de estos usuarios en relación con el proveedor.

3.7.8 Apoyo en la conversión de aplicaciones: Se define al apoyo que el proveedor puede proporcionar al cliente durante el proceso de conversión y traslado de las aplicaciones y programas al nuevo equipo. Por ejemplo convertir programas del lenguaje Cobol a Visual Basic. Obviamente este factor dependerá de la cantidad de programas que se convertirán a la nueva plataforma de hardware, como consecuencia del cambio de equipo. En la actualidad debido a las plataformas abiertas que se están utilizando, el apoyo de los proveedores en la conversión de aplicaciones ya no constituye un problema. Algunos conceptos a tomar en cuenta son los siguientes:

1. La experiencia que tiene el proveedor en el apoyo a otros clientes en conversiones requeridas.
2. Facilidad de iniciar la conversión de aplicaciones antes de la llegada del equipo, a fin de reducir el tiempo y costo de la operación paralela de ambos equipos.

3.8 Evaluación financiera de las propuestas

La evaluación financiera de las propuestas es el proceso mediante el cual el administrador del proyecto de cambio de equipo define y evalúa las características y los factores económicos de los equipos que deben considerar la empresa. El resultado de esta evaluación, sumando al resultado de la evaluación técnica, constituyen la plataforma de decisión del equipo y de la solución que se adquirirá.

145629

5.9 Alternativas de adquisición y financiamiento

A continuación se muestran diferentes alternativas de adquisición computacional que puede utilizarse con más frecuencia: renta del equipo, compra y arrendamiento financiero.

3.9.1 Renta: La renta es el proceso por el cual usuario toma en renta el equipo del proveedor por un periodo indefinido como obligatorio, y al terminar este período, suele tenerse tres alternativas:

1. Se debe cancelar el contrato y regresar el equipo al proveedor.
2. Renovar el período de renta, en cuyo caso es posible negociar con el proveedor de equipo un descuento sustancial en la nueva renta.
3. Ejercer la opción de compra del equipo. De hecho, en esta alternativa un porcentaje de las rentas pagadas al proveedor puedan aplicarse al pago al ejercer la opción de compra.

La renta de computadoras tiene algunas ventajas con respecto a otras alternativas de adquisición de equipo.

<u>Ventajas:</u>

1. No implica un desembolso inicial de dinero, lo que hace la operación cómoda desde el punto de vista financiero.

2. El proveedor de equipo computacional es responsable de mantenerlo funcionando en buen estado. De hecho, bajo esta alternativa se puede esperar un mejor servicio del proveedor.
3. No es una opción de compra obligatoria para el usuario en el largo plazo, lo cual permite tener flexibilidad en hacer cambios a la configuración seleccionada.
4. Evita la obsolencia y es más fácil hacer un cambio de equipo posterior, ya que no existe un compromiso a largo plazo del equipo.

Desventajas:

1. La renta usualmente es más costosa si el equipo seleccionado se empleara por un largo período.
2. Las rentas están sujetas a incremento por parte del proveedor, ya sea como un incremento real del precio o, bien, como consecuencia de devaluaciones de la moneda.

3.9.2 Compra: Una segunda opción es la compra del equipo computacional. Esta opción se hace cada vez más popular entre los usuarios que adquieren soluciones no propietarias de varios proveedores.

Las ventajas que presenta la adquisición del equipo son:

1. Es el método más barato cuando el equipo se requiere por períodos largos de tiempo.

2. No existen incrementos en los pagos y las devaluaciones de la moneda no afectan los flujos del negocio.
3. Se tiene un valor de recuperación del equipo al final del período.

Desventajas:

1. Más fácil llegar a tener un equipo obsoleto, debido al compromiso a largo que se adquiere con el equipo.
2. La solución adquirida no es la adecuada resulta el método más caro.
3. Implica un desembolso fuerte de dinero con el problema del financiamiento.
4. Se tiene incertidumbre en el valor de reventa del equipo, siendo muy frecuente el desuso total del mismo.
5. Se requiere tener una visión a largo plazo del negocio.

3.6.3 Arrendamiento financiero: Por último, el arrendamiento financiero es una alternativa viable para la adquisición de equipo computacional en la mayoría de los países. Presentes están algunas de las ventajas del esquema de renta, pero en el fondo se trata de una compra. En la mayoría de los casos la operación se realiza a través de un tercero, la arrendadora, la cual paga el equipo al proveedor y retiene la propiedad del equipo hasta que el usuario le liquida cada uno de los pagos. Es usual que el usuario pague al proveedor mensualmente el mantenimiento del equipo.

3.10 Modelo propuesto para estimar los requerimientos de equipo

Una vez que se han desarrollado los pasos previos, se puede iniciar el estudio de requerimientos para el nuevo equipo. Estimar el paso del procedimiento es quizás el más difícil y empírico de llevar a cabo, ya que no existe un método probado que garantice el éxito.

Los recursos que deben estimarse durante esta fase del proceso incluyen los siguientes:

- Capacidades de cómputo expresados, por ejemplo, en el número de instrucciones por segundo.
- Capacidades de almacenamiento en memoria principal expresadas, por ejemplo, en megabytes.
- Capacidades de almacenamiento secundario expresadas por ejemplo, en megabytes o gigabytes.
- Capacidad total de impresión requerida expresada, por ejemplo, en líneas de impresión.
- Cantidad de terminales requeridas para la captura o consulta de información.

- Hardware especializado que se requiera para llevar a cabo funciones especiales, tales como terminales inteligentes, concentradores, ruteadores, etc.

3.11 Requerimientos obligatorios y opcionales

Los requerimientos que se especificarán en el nuevo equipo pueden clasificarse como obligatorios y opcionales.

Requerimientos obligatorios

Se define como el conjunto de características que deben estar, de forma obligatoria y necesaria, presentes en el equipo o solución presentada por el proveedor, a fin de que no sea descartado de antemano. Ejemplos de éstos podrán ser los siguientes:

- El costo total del equipo o el presupuesto máximo autorizado.
- El tiempo máximo de entrega del equipo requerido.
- Compatibilidad con el lenguaje computacional actual, a fin de minimizar el esfuerzo de conversión de los programas al nuevo equipo.
- El apoyo del proveedor durante la conversión de las aplicaciones.

La definición adecuada de restricciones en el proceso de cambio de equipo facilitará el trabajo durante la evaluación técnica y financiera de las propuestas, ya que solamente se analizarán aquellas propuestas o soluciones que hayan cumplido con todos los requerimientos obligatorios.

Obviamente, el cálculo no considera situaciones que puedan presentarse durante el horizonte de planeación, como las siguientes, lo cual puede traer como consecuencia desviaciones en el cálculo

- Crecimiento del negocio por arriba o por abajo del porcentaje estimado.
- Fusiones o compra de nuevos negocios.
- Rediseño o cambios importantes en las aplicaciones.

3.13 Requerimientos Futuros de las nuevas aplicaciones

En esta parte del modelo propuesto se calculan los requerimientos que tendrá cada uno de los recursos que están siendo estimados, considerando las aplicaciones que se desarrollarán durante el horizonte de planeación del proyecto.

La manera de llevar a cabo lo anterior es conocer las aplicaciones que van a ser desarrolladas durante el horizonte de planeación, tal como se sugiere en el plan de aplicaciones. Para cada una de las aplicaciones habrá que elaborar un prediseño estimando los volúmenes requeridos de cada uno de los recursos. El valor o los valores que se obtengan de este análisis, deberán de sumarse a los valores obtenidos en la sección anterior, de tal forma que podamos estimar el valor total de los requerimientos del nuevo equipo para cada uno de los recursos.

A continuación se presenta un ejemplo sencillo de la aplicación del modelo:

Suponga que se desea estimar el área en disco a considerar en el nuevo equipo computacional. Las aplicaciones actuales ocupan un total de 792 Mb. El crecimiento del negocio se estima en un 10% anual en un horizonte de planeación de 5 años. Los resultados son los siguientes:

RFA = RAA * (1 + PC/100) HP

= (792)* (1 + 10/100) 5

= 792 * (1.1)

= 1275.5 Mb.

Además, se estiman que los requerimientos futuros de las aplicaciones nuevas (RFN) serán de 214 Mb. , Considerando los nuevos sistemas por desarrollar. Así, los requerimientos totales del nuevo equipo se calcularán de la siguiente manera:

= RFA + RFN

= 1275.5 + 214

= 1489.5 Mb.

Lo anterior quiere decir que el nuevo equipo deberá tener una capacidad de almacenamiento de 1489.5 Mb. Para soportar los requerimientos actuales y futuros de los próximos 5 años.

3.13 Evaluación técnica de las propuestas

La evaluación técnica de las propuestas es el proceso mediante el cual el administrador del proyecto de cambio define y evalúa las características y los factores técnicos de los equipos a considerar. El resultado de esta evaluación técnica, sumado al resultado de la evaluación económica y financiera, constituye la plataforma de decisión del equipo y de la solución a adquirir.

En la siguiente sección se analizarán las actividades a desarrollar durante la evaluación económica y financiera.

3.14 Del RFP (Request For Proposal)

La evaluación técnica de los equipos se facilita con la correcta elaboración de las propuestas por parte de los proveedores de soluciones.

Una vez que se ha terminado la estimación de los requerimientos para el equipo nuevo que se va a adquirir, es necesario elaborar una solicitud o requisición de propuesta. La requisición de propuesta (RFP) es un documento que define los requerimientos de la organización para el equipo o la red requerida por la organización.

- Sirve como una propuesta del sistema que invita a los proveedores a participar en el concurso.
- Establece los primeros puntos de evaluación y negociación entre los proveedores de soluciones computacionales y la organización.
- Obliga al administrador del proyecto de cambio de equipo a formalizar el proceso de determinación de los requerimientos de equipo.
- Constituye un documento que describe claramente las prioridades técnicas del sistema.

La estructura del RFP puede variar dependiendo de la naturaleza del proyecto de cambio de equipo que se desarrollará. No obstante lo anterior se recomienda incluir, entre otros, los siguientes puntos:

Introducción:

En esta sección se puede integrar la siguiente información:

- Datos generales del responsable del proyecto.
- Fecha límite para recibir la propuesta por parte del proveedor de equipo computacional que desea concursar.
- Fecha límite para realizar por parte del proveedor las presentaciones y/o demostraciones del equipo propuesto.

- Bases y lineamientos generales que serán utilizados para hacer la comparación entre los diferentes equipos.
- Breve descripción de la situación actual de la compañía y de la función de informática dentro de la empresa.

3.15 Requerimientos del sistema computacional

En esta sección se puede incluir la siguiente información:

- Requerimientos del equipo actual vs. El equipo propuesto.
- Requerimientos obligatorios y opcionales, tales como sugiere en la sección de determinación de requerimientos.

- Información detallada de las pruebas de benchmark o pruebas de rendimiento, que serán efectuadas a las soluciones propuestas que estarán concursando.
- Y en términos generales deberá incluirse toda la información relevante descrita en la sección de determinación de requerimientos.

PRINCIPALES SOLUCIONES DE UN PROVEEDOR A LOS PROBLEMAS EN LA VENTA

4.1 El costo de una nueva adquisición de un producto o servicio

Es el costo que representa para una empresa, negocio o establecimiento la adquisición de sus compras rutinarias (lo que siempre compran); la adquisición de un producto alternativo o sustituto y de nuevos productos o servicios.

Tendremos que definir lo que el cliente busca en nosotros. Si tenemos lo fundamental que es: la entrega, precios, calidad, tiempo de venderle, empatía, tradición, amistad, etc., habrá que preguntarnos: ¿Por cuáles de estos motivos me compran a mí o a mi empresa?

Debemos tomar en cuenta que conocer a nuestro cliente a fondo implica principalmente saber si le vendemos a subdistribuidores, mayoristas o detallistas, y a qué mercado están dirigidos o qué tipo de clientes les compran y saber las características que poseen sus compradores o consumidores. En caso de que nosotros le vendamos al público en general, debemos conocer las características

de nuestros compradores o consumidores. Entonces debemos tener cuidado con el tipo de producto que ofrecemos. Estos pueden ser:

- **Producto alternativo**: Es ofrecerle el mismo producto pero de diferente marca y satisfacer la misma necesidad. Es un producto de la competencia directa.

Este es uno de los grandes problemas cuando se realiza una venta, sea de lo que siempre nos compran o de un nuevo producto, desconocemos el mercado de nuestro cliente, su tipo de comprador o lo que realmente busca. Su motivo de compra, adquisición, uso o para qué lo va utilizar o necesitar.

- **El producto sustituto**: Es ofrecer un producto diferente, que cumpla la misma necesidad, pero que posee otros atributos.

Si queremos crecer, debemos ofrecer diferentes productos o servicios, de igual o diferente marca, pero cuando menos con las mismas condiciones que los que veníamos ofreciendo; sea éste alternativo, sustituto o un nuevo producto.

4.2 La gran información existente de muchos productos o servicios.

El problema fundamental cuando nos enfrentamos a realizar una compra es el de no decidir acerca de las bondades de un producto o servicio. Esto lo atribuimos principalmente a estos aspectos:

- Exceso de productos similares
- Saturación de información por parte de los oferentes de productos
- Capacidad de discernir por parte del comprador
- Temor a la decisión

Exceso de productos similares: Debido al libre comercio existente en el mundo, cada vez se generan (hasta cierto punto) nuevos productos, mejoras a los productos o conjunción de ambos en uno nuevo. Lo que hace que sea muy difícil la decisión por parte del comprador y un amplio manejo de cualquier producto por parte del asesor de la venta.

Es indispensable que además de conocer a fondo sus productos, debe ofrecer un "traje a la medida" para cada cliente. La venta que se realiza tiene las siguientes características:

a) Se debe ofrecer una solución
b) Plantear los usos o ventajas de lo que se vende
c) Saber explicar las diferencias de cada producto desde el punto de vista del consumidor, comprador y usuario.

Por lo que la venta cada vez se vuelve más compleja, ya que es sólo el que paga o toma la decisión, sino el que además usa el producto o servicio y el que decide la compra de éste.

4.3 Saturación de información por parte de los oferentes de productos

Es importante que no envíe comunicación acerca de lo que su negocio vende a otras empresas, cada vez éstas son menos importantes. Vivimos ante una saturación de información que hace que los destinatarios no la revisen o ni siquiera lleguen a sus manos.

Cada vez la venta se hace más personal y al mismo tiempo más cálida; ya no es tanto la buena relación, sino el conocimiento, ventajas, usos, aplicaciones y principalmente. Lo que nuestro comprador gana con compramos a nosotros.

Si lo que uno desea es promocionarse por este medio, le sugiero los siguientes pasos:

1. Contacte a la empresa y pregunte a quién le puede enviar la información, hágalo llagar a varias personas (determine con la información de la empresa y con su experiencia quién decide, quién hace el pedido, quién usa y quién recomienda).
2. Confirme que haya llegado la información a las personas antes mencionadas.
3. Realice una cita para hacer una presentación más amplia. Si es posible, en primer lugar, con el usuario del producto o servicio que ofrece.

4.4 Capacidad para discernir por parte del comprador

Por las razones antes mencionadas, es muy difícil que el comprador pueda tomar la decisión rápidamente, pues tiene que establecer relaciones internas con quienes autorizan el pedido, lo envían o de quienes lo utilizan.

Es fundamental que nosotros también hagamos la labor de asesoría de venta con el comprador (lo que se busca es la recompra o la lealtad), no sólo para que conozca nuestro producto, sino para elevar su autoconfianza, explicándole fundamentalmente nuestro precio y el motivo de éste (sea caro, barato o competitivo), las condiciones de entrega y las razones por las cuales nos escogieron a nosotros como proveedores.

Recuérdese, lo que se persigue es que cuando al comprador le soliciten el mismo o parecido material o servicio, siempre recurra a nuestro negocio.

4.5 Temor a la decisión

Desgraciadamente todavía, la mayoría de las empresas, consideran que es correcto el refrán que dice: "más vale malo por conocido, que bueno por conocer", esto nos hace más difícil la labor de convencimiento para nuestro trabajo.

Habrá que tomar en consideración que para cada negocio existe un arma fundamental para lograr una venta. En ferreterías, es el tiempo de entrega, flete y

precio, en impresión es precio, tiempo de entrega y calidad. En bienes e inmuebles son las ventajas para la familia y en autos de uso y la ubicación del comprador dentro de la familia o en la empresa.

Por lo que es imprescindible que la relación entre comprador - vendedor sea armónica y no de conflicto o de superioridad por parte de quien ofrece un servicio o producto.

4.6 El grado de involucramiento del vendedor – comprador

Lo abarcaremos en 2 aspectos: las relaciones comprador - vendedor y las relaciones vendedor - comprador.

Habrá que tomar en cuenta 2 consideraciones propias de esta relación, el comprador.

a) Posee muchas predisposiciones ante un vendedor: actitudes, creencias, valores y metas. Estas son creadas porque espera de un vendedor la satisfacción de la necesidad, lograr mantenerse en su puesto, conseguir el mejor precio, obtener algo más que de lo que ya posee, etc.
b) Considera al vendedor ya no hablador, tolerante, competitivo, optimista y nervioso; sino un manipulador. De ahí que el asesor de ventas, representante técnico o vendedor debe tener presente, para lograr un pedido, los siguientes aspectos:

1. **Se debe lograr la entrevista con el comprador.** La primera impresión es fundamental, ésta empieza desde la forma en que se solicita la cita. No sólo que realice ésta la secretaria, sino también el asesor.

2. **Presentación impecable.** El vendedor debe considerar que la mayoría de las veces la cita nos la otorgan cuando:

a) Existe poco trabajo
b) Están buscando nuevos proveedores
c) Alguien nos recomendó
d) Nos recuerdan por nuestra publicidad

Es fundamental que sepamos distinguir estas diferencias y determinar el motivo por el cual nos otorgan la cita, pero en caso de ser por los 2 primeros motivos habrá que hacer énfasis al comprador, de quiénes son nuestros clientes y la publicidad o promoción que manejamos. No sólo dejar al prospecto una cotización, sino folletos, muestras de otros trabajos (si puede), curriculum de la empresa, algo que le indique porqué le conviene compramos a nosotros y no a la competencia o a otros proveedores.

3. **Determinar qué parte de la presentación le llamó la atención.** Sólo de esta manera podremos determinar cuáles son aquellos elementos que le interesan de nosotros: calidad, tiempo de entrega, precio, servicio, presentación,

garantía, empaque, publicidad, cuentas que manejamos, entre otras las más importantes.

Se tiene que recordar que un comprador nos hablará cuando nos requiera siempre y cuando:

a) Recuerde la máxima información posible de nosotros
b) La forma en que hicimos la presentación de nuestro producto o servicio

Por otro lado, existen básicamente 3 tipos de compra:

1. **La recompensa directa**. Es cuando ya se tiene al proveedor y sólo se le envía el pedido. Es lo que todos buscamos, pero a la larga nos puede sacar del mercado si no mejoramos o buscamos ventajas competitivas. Aquí la compra que domina es la base de alianzas estratégicas y de amistad, pero puede convertirse en sólo una fría relación.

2. **La recompensa modificada**. Es la que siempre nos ha comprado, pero habrá que modificar nuestro producto o servicio, de acuerdo con las necesidades. La relación es con base en amistad y alianza, pero puede caer en una vil transacción, si no nos adecuamos al cliente.

Es el caso de cuando nos llama un cliente y nos dice: "necesito que venga, pues quiero hacer un pedido diferente a lo que les compro...", o "requiero esto que

siempre me surten, pero con algunas modificaciones...", sea en tiempo de entrega, precio por volumen o un producto o servicio adaptado a alguna necesidad.

3. Nueva compra. Esta se divide en 3 aspectos que debe cuidar el asesor:

- Duda sobre las necesidades. Cuando el comprador desconoce específicamente las características del producto o servicio que requiere. En este caso. Es fundamental la cultura al cliente a través de muestras de lo que hemos realizado, ayudarle a decidir con base a nuestra experiencia, explicarle las ventajas, usos, desventajas, precio, condiciones de entrega, etc., de varias opciones de productos o servicios que le ofrezcamos.

- Es cuando lanzamos al mercado un nuevo producto o servicio, o el comprador desconoce lo que le ofrecemos.
- Dudas sobre el mercado. Es cuando existen muchos vendedores sobre el mismo producto y es imposible que el comprador pueda adquirir entre las diferentes y muchas propuestas. Para este caso, además de la estrategia de cultura al cliente, es importante vendedor testimoniales (hacerles saber quiénes son nuestros clientes) pero, no tanto a los que nos compran, sino el tiempo que tienen de ser nuestros clientes. Así como mostrar lo que hemos logrado con cada uno de éstos.

- Dudas sobre la transacción. Aquí nos referimos a la forma en que se levan a cabo las condiciones de venta, el tiempo de entrega y la calidad. Lo que importa en este punto (que es el cierre de la venta) es especificar claramente la fecha de entrega al cliente, el momento y el lugar que se indique.

Esta situación es muy frecuente entre las empresas que desean abrir mercado, lo primero, que los compradores "repelan" no es el precio o la calidad del producto, sino la entrega a tiempo o cuando lo prometen. De aquí que el tiempo de entrega sea un arma competitiva y determinante.

4.7 Grado de involucramiento comprador – vendedor

Antes de explicar las principales formas para lograr una venta, es fundamental que el vendedor esté consciente de que lo importante no es ganar una discusión, sino lograr la venta a través de una estrategia de flanqueo (de lado). No atacar al cliente o tratarlo de convencer a la fuerza.

Por lo tanto, es requisito indispensable que el vendedor esté consciente que su presentación personal, la del producto o servicio que ofrece debe ser la más adecuada; y además para llevar a cabo una venta tiene que considerar:

a) Resolver problemas de la entrega, distribución, condiciones de pago y calidad del producto.
b) Requiere controlar la tensión y ansiedad que a veces se refleja por el exceso de trabajo o por la premura de cerrar el pedido.
c) Es importante que para lograr tanto una cita como un posible pedido cuidar la reputación de la empresa.

Al mencionar la imagen y la presentación del producto o servicio no necesariamente nos referimos a que existen folletos, publicidad, muestras, etc., sino la forma de hacerla, Ya que es fundamental que el comprador nos entienda, tanto sobre las ventajas del producto como sus usos, aplicaciones, restricciones y condiciones de venta.

El asesor o representante técnico (mal llamado vendedor, promotor o ejecutivo de ventas) ya no es aquel que manipula, persuade o establece un compromiso para lograr una venta.

Si el asesor utiliza como una técnica fundamental en una venta la persuasión se enfrenta a estos problemas:

1. Una forma antigua de hacer negocios. Esto resulta en los años 70, "cómpreme porque va a subir", "es una gran oportunidad", "pronto se va a acabar", "que pero le pone", etc.
2. Esta forma utiliza un modelo ganar - perder, donde el perdedor es el comprador, y por lo tanto;

3. Son relaciones a corto plazo, es muy difícil que me vuelva a comprar.

Si su técnica es la del compromiso, el que le compran, ya sea por "amistad". "Porque lo conocen", "porque les he resuelto problemas", "porque siempre le han comprado", etc. :

1. Ambos pierden, ya sea que uno engaña y el otro siente que debe comprar por el compromiso.
2. También es una relación a corto plazo.

Otra técnica mal empleada, que es la que actualmente aplican la mayoría de las empresas, es la de satisfacción del cliente.

La satisfacción de éste ya es tan importante como en la década pasada, sino (aunque no les guste) es una relación en que no se tiene control de ella. No se sabe qué va a pedir el cliente, qué requerirá; y honestamente, luego nos enfrentamos con cada cliente, que no sabemos qué cosa rara o imposible nos va a solicitar.

a) El asesor es un resolvedor de problemas, no de cultura del cliente.
b) Es una relación frágil, pues el día que no se da el servicio, se pierde al cliente.
c) El comprador siempre tendrá el poder de la relación.
d) Este tipo de relaciones es a mediano plazo o hasta que el proveedor ya no pueda cumplir.

Desgraciadamente, en nuestro medio, todavía existen empresas que sólo les interesa el tiempo de entrega, el precio más bajo en el mercado y la calidad con que se le está otorgando.

Recordemos que en días pasados una persona nos preguntó qué le sugeríamos ante un mercado en donde la competencia la acosa a través de sus clientes, puesto que sus competidores "ofrecen dinero a los compradores", y hasta reclaman "sólo le compran a esta persona, si tienen mejor precio que esa empresa".

Hemos comentado que esto era una situación de mercado, que era normal (excepto el ofrecer dinero, que considero que es una situación que poco a poco está cambiando). Que su éxito estaba en la relación que establecía con el cliente; en donde además de no "chotear" el precio del mercado ofrecía a sus compradores una negociación, reciprocidad y una alianza con respecto a lo que les estaba ofreciendo. Lo que los compradores llaman regateo, pero no es precio, sino en condiciones de pago, calidad, entrega, características, etc.

El regateo es el paso previo a la negociación, es un concepto más analítico en la compra (generalmente se da en la transmisión de la primera venta o pedido)

Aquí surge:

- La comparación con la competencia.
- La cooperación de ambas partes.
- La competencia mediante el comportamiento del vendedor (al servicio que se le ofrezca y sea mejor que los otros).
- Es un juego donde ambos buscan ganar algo; pero a veces sólo gana el comprador al no comprar, aunque la ventaja es que conoce el producto o servicio ofrecido. Y gana el vendedor cuando ve una posibilidad de venta.

En este último punto habrá que tener mucho cuidado, ya que el comprador esté convencido, generalmente no se levanta un pedido en la primera visita.

Después de este paso, sigue en la negociación la estrategia más recomendable y la mejor para crecer, pero si no cuenta con la infraestructura suficiente o conoce su capacidad instalada para atender a diferentes clientes es contraproducente. La negociación se caracteriza por:

- Se reconoce al comprador como un individuo con decisión y a través de su comportamiento puede aumentar la relación vendedor - comprador.
- Existe un clima de confianza entre ambos.
- El papel de asesor realmente es de consejero y asesor, no de rival.
- Existe un incremento de información para fomentar loa relación a largo plazo.
- Es un modelo ganar - ganar.

Siguiendo con el comentario de esta persona, además de hacer lo anterior, ha establecido con sus compradores una relación de reciprocidad, por lo que la mayoría de los asesores deben de llevar a cabo este tipo de relación con el mayor número de clientes posible. Esta tiene las siguientes cualidades:

- Se establece un patrocinio mutuo.
- Están implicados los valores de justicia, utilidad y buena fe.
- Se crece con los clientes.
- La relación es de amistad y cooperación mutua.

4.8 Los valores agregados, las ventajas comparativas o competitivas.

Es fundamental este último punto, pues los valores agregados están caducos. Actualmente, muchos desean competir con base en éste: algo que doy de más por el mismo precio. La mayoría de las compañías están ofreciendo esto, como entrega sin costo, estacionamiento, mismas condiciones de pago, etc. Esta forma de competir está empezando a pasar de moda.

Ahora todas las empresas ofrecen estos servicios, por lo que el competir con valores agregados es algo ya muy común.

Lo que todo empresario debe hacer es preocuparse si no cuenta con dichos valores agregados, porque en lugar de contar con las armas para competir,

tenemos desventajas competitivas. También cuídese cuando dentro de los valores agregados le piden algo que está fuera de su negocio, como hacer una tarima especial y que implica mucho gasto; debe hacer sólo un empaque; vender algo que no está dentro de su línea, entre otros; porque lo que pueden provocarle sus clientes es:

1. Sacarlos de su verdadero negocio. Que implica el vender más aquello que para mi negocio es servicio, productos con costo muy altos; difíciles de desplazar o conseguir.
2. Ser su resolvedor de problemas y el día que no pueda solucionar el problema, difícilmente le volverá a comprar.

Lo que actualmente está de moda, es el ofrecer lo que se llama valor aumentado, que es dar información al cliente de nuestro producto. No es propiamente lo que denominamos ingeniería del valor (explicar al cliente aquello que no ve en mi producto), sino hacer énfasis en el servicio, las condiciones de entrega, el material, la garantía, el uso, etc.

Si vendemos muebles de madera es fundamental explicarle al cliente que no estamos en contra de la ecología, la duración, a que es resistente, el tipo de madera, cómo debe limpiarse, las ventajas sobre otro tipo de muebles y si es posible vender los aditamentos para limpiarlos.

Al lograr manejar a fondo estas situaciones, estaremos aplicando una comercialización adaptada al cliente, que implica conocer las capacidades de mi

empresa para hacer las cosas y por otro lado conocer también las necesidades del cliente.

Teniendo lo anterior, estaremos en posición de poder competir por aquellas que se llaman ventajas comparativas y ventajas competitivas.

Las primeras consisten en aquello que "doy mejor servicio que la competencia"; mientras que las segundas son aquellas que "ofrezco que nadie más puede hacerlo".

Las ventajas comparativas (básica en el cierre de ventas), son las que nuestros clientes buscan y son la rapidez, pronta entrega, menor pérdida de tiempo para la decisión de compra, cercanía, la cualidad fundamental de nuestro producto; entre otras más que sobresalen.

Estas habrán de manejarse cuando nos enfrentemos a un cliente que tiene otras cotizaciones o puede adquirir el mismo o similar producto en otra empresa o establecimiento.

Las ventajas competitivas son cuando realmente logramos ser los únicos proveedores de un cliente o cuando sólo nosotros contamos con el producto o servicio. En el actual mercado en el que vivimos es muy difícil ser los únicos, debido a que cada vez los productos o servicios son más similares, o existen

aquellos que cubren las mismas necesidades pero son diferentes a los que nosotros ofrecemos (productos alternativo o sustituto).

En este caso, lo que se recomienda para competir es además de lograr estas ventajas, es:

- Se debe innovar los productos o servicios; con relación a calidad, presentación, usos y resistencia.
- Se debe llevar el actual producto o servicio a otros mercados.
- Se debe conseguir hacer trajes a la medida del producto o servicio, del cual somos los únicos en el mercado, único proveedor, mejor proveedor o alternativa.

Finalmente, la confiabilidad con que usted entregue su producto o servicio en lo que respecta a mantener la calidad, tiempo de entrega, claras especificaciones y precio competitivo será el éxito de su negocio.

SELECCIÓN DEL SOFTWARE

5.1 La panoramica del Software

Durante las tres primeras décadas de la informática, el principal desafío era el desarrollo del hardware de las computadoras, de forma que se redujera el costo de procesamiento y almacenamiento de datos. A lo largo de la década de los ochenta, los avances en tecnoelectrónica han dado como resultado una mayor potencia de cálculo a la vez que una reducción del costo. Hoy, el problema es diferente. El principal desafío es mejorar la calidad , durabilidad y reducir el costo de las soluciones basadas en computadoras (soluciones que se implementan vía software). La capacidad de las grandes computadoras de la era de los ochenta está hoy disponible en una computadora personal. Las enormes capacidades de procesamiento y almacenamiento del hardware moderno representan un gran potencial de cálculo. El software es el mecanismo que nos facilita utilizar y explotar este potencial.

5.1.1. Proceso evolutivo del software

El contexto en el que se ha desarrollado el software está fuertemente ligado a las casi cinco décadas de evolución de los sistemas informáticos. Un mejor rendimiento del hardware, una reducción del tamaño y un costo más bajo, han dado lugar a sistemas informáticos más sofisticados. Hemos pasado de los

procesadores con bulbos a los dispositivos microelectrónicos que son capaces de procesar 200 millones de instrucciones por segundo. Se habla de una "nueva revolución industrial"; a la llegada de estos componentes la "tercera ola de cambio" en la historia de la humanidad; de forma que la transformación de la sociedad industrial en una "sociedad de la información" tendría un profundo impacto en nuestras vidas. Por lo que la información y el conocimiento (controlados por computadoras) serían el foco principal en el presente siglo, la "comunidad electrónica" creada mediante software y redes de información es la clave para el intercambio del conocimiento alrededor del mundo. Al comenzar los años 90, se describió un "cambio en el poder" en el cual las antiguas estructuras de poder (gubernamentales, educativas, industriales, económicas y militares) se desintegrarían a medida que las computadoras y el software nos llevaran a la "democratización del conocimiento".

Se describe la evolución del software dentro del contexto de las áreas de aplicación de los sistemas basados en computadoras. Durante los primeros años de desarrollo de las computadoras, el hardware sufrió continuos cambios' mientras que el software se contemplaba simplemente como un añadido' La programación de computadoras era un arte donde existían pocos métodos sistemáticos. El desarrollo del software se realizaba virtualmente sin ninguna planificación hasta que los planes comenzaron a descalabrarse y los costos a crecer. Durante este período se utilizaba en la mayoría de los sistemas una orientación por lotes.

Durante los primeros años, lo normal era que el hardware fuera de propósito general. Por otra parte, el software se diseñaba a medida para cada aplicación y tenía una distribución relativamente pequeña. El software como producto (es decir, programas desarrollados para ser vendidos a uno o más clientes) estaba en su infancia. La mayoría del software se desarrollaba y era utilizado por la misma persona u organización. La misma persona lo escribía, lo ejecutaba y, si fallaba, lo depuraba. Debido a que la movilidad en el trabajo era baja, los ejecutivos estaban seguros de que esa persona estaría allí cuando se encontrara algún error.

Debido a este entorno personalizado del software, el diseño era un proceso implícito, realizado en la mente de alguien, y la documentación normalmente no existía. A lo largo de los primeros años aprendimos mucho sobre la implementación de sistemas informáticos, pero relativamente poco sobre la ingeniería de las computadoras. Sin embargo, en honor a la verdad, debemos reconocer que durante esa era se desarrollaron muchos sistemas informáticos excepcionales. Algunos de ellos todavía se siguen usando hoy y, por sus características, siguen siendo admirados con toda justicia.

La segunda era en la evolución de los sistemas de computadora se extiende desde la mitad de la década de los 60 hasta finales de los setenta. La multiprogramación y los sistemas multiusuario introdujeron nuevos conceptos de interacción hombre-máquina. Las técnicas interactivas abrieron un nuevo mundo de aplicaciones y nuevos niveles de sofisticación del hardware y del software. Los sistemas de tiempo real podían recoger, analizar y transformar datos de múltiples fuentes, controlando así los procesos y produciendo salidas en milisegundos en

lugar de en minutos. Los avances en los dispositivos de almacenamiento en línea condujeron a la primera generación de sistemas manejadores de bases de datos.

La segunda era se caracterizó también por el establecimiento del software como producto y la llegada de las "casas de software". El software ya se desarrollaba para tener una amplia distribución en un mercado diversificado. Los programas se distribuían para computadoras grandes y para minicomputadoras, a cientos e incluso a miles de usuarios. La industria, el gobierno y la universidad se aprestaban a "desarrollar el mejor paquete de software".

Conforme crecía el número de sistemas informáticos, comenzaron a extenderse las bibliotecas de software de computadora. Las casas desarrollaban proyectos en los que se producían programas de decenas de miles de sentencias. Los productos de software comprados al exterior incorporaban cientos de miles de nuevas sentencias. Una nube negra apareció en el horizonte. Todos esos programas todas esas sentencias fuente tenían que ser corregidos cuando se detectaban fallos, modificados cuando cambiaban los requisitos de los usuarios o adaptados a nuevos dispositivos hardware que se hubieran adquirido. Estas actividades se llamaron colectivamente *mantenimiento del software*. El esfuerzo gastado en el mantenimiento del software comenzó a absorber recursos en una medida alarmante.

Aún peor, la naturaleza personalizado de muchos programas los hacía virtualmente imposibles de mantener. Había comenzado una "crisis del software".

La tercera era en la evolución de los sistemas de computadora comenzó a mediados de los setenta y llega hasta el momento actual. El procesamiento distribuido múltiples computadoras, cada una ejecutando funciones concurrentemente y comunicándose con alguna otra incremento notablemente la complejidad de los sistemas informáticos. Las redes de área local y de área global, las comunicaciones digitales de alto ancho de banda y la creciente demanda de acceso "Instantáneo" a los datos, supusieron una fuerte presión sobre los desarrolladores del software.

La tercera era también se caracteriza por la llegada Y el amplio uso de los microprocesadores y las computadoras personales. El microprocesador es una parte integral de un amplio espectro de productos "Inteligentes" que incluyen automóviles, hornos microondas, robots industriales y equipos de diagnóstico de suero sanguíneo. En muchos casos, la tecnología del software es integrada en esos productos por equipos técnicos que conocen el hardware, pero que a menudo, no tienen experiencia en desarrollo de software.

Las computadoras personales han sido el catalizador del gran crecimiento de muchas compañías de software. Mientras que las compañías de software de la segunda era vendían cientos o miles de copias de sus programas. las compañías de software de la tercera era venden decenas e incluso centenares de miles de copias. El hardware de las computadoras personales se ha convertido rápidamente en un producto estándar, mientras que el software que se suministre con ese hardware, es lo que marca la diferencia. De hecho, mientras que las

ventas de computadoras personales se estabilizaron hacia la mitad de los 80, las ventas de productos de software han continuado creciendo. Mucha gente en el campo industrial y muchos particulares han gastado más dinero en software que lo que se gastaron en la computadora sobre la que se ejecuta el software.

La cuarta era del software de computadora está empezando ahora. Las tecnologías orientadas a los objetos están desplazando rápidamente a enfoques de desarrollo de software más convencionales en muchas áreas de aplicación. predijeron que las computadoras de la "quinta generación", arquitecturas de cálculo totalmente diferentes y su correspondiente software tendrá un profundo impacto en el equilibrio de la potencia política e industrial de todo el mundo. Las técnicas de cuarta generación para el desarrollo de software están cambiando la forma en que algunos segmentos de la comunidad informática construyen los programas de computadora. Por fin, los sistemas expertos y el software de inteligencia artificial se han trasladado del laboratorio a las aplicaciones prácticas, para un amplio rango de problemas del mundo real. El software de redes neuronales artificiales ha abierto excitantes posibilidades para el reconocimiento de formas y habilidades de procesamiento de información al estilo de como lo hacen los humanos.

Conforme nos movemos en la cuarta era, continúan intensificándose los problemas asociados con el software de computadoras:

1. La sofisticación del hardware ha dejado desfasado nuestra capacidad de construir software que pueda explotar el potencial del hardware.
2. Nuestra capacidad de construir nuevos programas no puede dar abasto a la demanda de nuevos programas.
3. Nuestra capacidad de mantener los programas existentes está amenazada por el mal diseño y el uso de recursos inadecuados.

5.1.2. Explosividad del software

En los primeros días de la informática, los sistemas basados en computadora se desarrollaban usando técnicas de programación orientadas al hardware. Los programadores del proyecto se centraban en el hardware, debido a que era el factor principal del presupuesto en el desarrollo del sistema. Para controlar los costos del hardware. los programadores instituyeron controles formales y estándares técnicos. Exigían un análisis y diseño completo antes de que algo se construyera. Medían el proceso para determinar dónde podían hacerse mejoras. Dicho sencillamente, aplicaban los controles, métodos Y herramientas que reconocemos como ingeniería del hardware. Desgraciadamente, el software no era normalmente más que un añadido.

En los primeros días, la programación se veía como un "arte". Existían pocos métodos formales, v pocas personas los usaban. El programador aprendía normalmente su oficio mediante prueba y error. La función y los desafíos de la construcción del software de computadoras crearon un "orden" en la que pocos

se preocuparon por estandarizar . ¡El mundo del software era virtualmente sin disciplina y muchos principiantes lo explotaban!

Hoy, la distribución de costos en el desarrollo de sistemas informáticos ha cambiado drásticamente. El software, en lugar del hardware, es normalmente el elemento principal del costo. En las pasadas décadas los desarrolladores y técnicos se habían hecho las siguientes preguntas:

* ¿Por qué lleva tanto tiempo terminar los programas"
* ¿Por qué es tan elevado el costo?
* ¿Por qué no podemos encontrar todos los errores antes de entregar el software a nuestros clientes?
* ¿Por qué nos resulta difícil constatar el progreso conforme se desarrolla el software?

Estas y muchas otras preguntas son una manifestación del carácter del software y de la forma en que se desarrolla un problema que ha llevado a la adopción de la ingeniería del software como práctica.

5.1.3. Obsolescencia de software

En los años cincuenta y sesenta, muchos comentaristas especializados criticaron duramente a la industria del metal por la falta de inversión en las fábricas. Las fábricas habían comenzado a deteriorarse; raramente se aplicaban los métodos de producción modernos; la calidad quedaba en entredicho y sin embargo el costo del producto final subía; y la competencia comenzó a ganarse una parte sustancial del

mercado. La dirección de esas industrias decidió no realizar la inversión de capital que se requería para mantenerse competitivas en sus entornos comerciales. La industria del metal sufrió esa situación durante mucho tiempo, perdiendo una significativa parte del mercado, beneficiando a la competencia extranjera competencia que tenía fábricas más modernas, usaba tecnología más moderna y obtenía ayudas de sus gobiernos para hacerlas extremadamente competitivas en los costos.

Durante ese período, muchos de nosotros, inmersos en la floreciente industria de las computadoras. Miramos a la industria del metal con desdén. "Si no están dispuestos a invertir en su propio negocio". Dijimos. "Se merecen perder parte del mercado". Esas palabras pueden ahora volverse contra nosotros.

Aún a riesgo de que suene un poco melodramático. El hecho es que la industria del software hoy está en una situación que es bastante similar a la de la industria del metal en los años cincuenta y sesenta. Tanto en las grandes empresas como en las pequeñas, lo que tenemos es una "fábrica de software" que envejece hay cientos de aplicaciones basadas en software en una situación critica y necesitan ser renovadas urgentemente:

- Las aplicaciones de sistemas de información escritas, hace veinte años que han sufrido cuarenta generaciones de cambios y que ahora son virtualmente imposibles de mantener. Incluso la más pequeña modificación puede hacer que falle todo el sistema.

- Las aplicaciones de ingeniería que se utilizan para generar datos críticos de diseño y que, sin embargo, a pesar de su edad y estado de conservación, realmente no se entienden. Nadie tiene un conocimiento detallado sobre la estructura interna de esos programas.
- Sistemas preinstalados (usados para controlar plantas de potencia, tráfico aéreo y fábricas, entre sus cientos de aplicaciones) que parecen extraños Y a veces tienen un comportamiento inexplicable, pero que no se pueden poner fuera de servicio porque no hay nada disponible para reemplazarlas.

No será suficiente "reparar" lo que está mal y dar una imagen moderna a estas aplicaciones. Muchos componentes de la fábrica de software requieren una reingeniería o reestructuración importante o, de lo contrario, no serán competitivos durante los años noventa. Desafortunadamente. Muchos directores de empresas parecen poco dispuestos a comprometer los recursos necesarios para emprender este esfuerzo de reestructuración. "Las aplicaciones todavía funcionan", dicen, "y no es 'económico' comprometer recursos para mejorarlos".

Alguien podría argumentar que la metáfora anterior no tiene fundamento. ¿De dónde vendría la competencia?", Se preguntan.

El hecho es que la competencia se está formando justamente ahora. Será amplia, será competente y estará aquí antes de lo que se piensa. Países del lejano oriente (Japón, Corea, Singapur), de Asia (India, China) y de Europa ofrecen un amplio espectro de profesionales altamente motivados, educados de

forma competente y a un costo relativamente bajo. Esa mano de obra se moverá rápidamente para adoptar los métodos y las herramientas de ingeniería del software más innovadores y puede muy bien llegar a ser una fuerza con la que ha, a que contar durante los últimos años de la década de los noventa.

Algunas empresas han tirado la toalla Y han empezado a "subcontratar". La empresa reduce al mínimo su personal dedicado a los sistemas de información y contrata o pacta con una tercera parte la gestión de todo el desarrollo del nuevo software, gran parte del mantenimiento de sistemas en marcha Y todas sus operaciones de computadora. Hasta este momento, la subcontratación está normalmente restringida a oficinas de servicio local.

Cierto, ¿qué pasará con el mañana? El hardware de computadoras está convirtiéndose ya en algo cómodo, disponible por muchos medios. Pero el software sigue siendo una industria en la que Estados Unidos ha sido un país "innovador, vital, fructífero'. ¿Mantendremos nuestra situación en lo más alto?

Algunos sabios opinan que existen razones económicas, políticas, tecnológicas y de seguridad nacional por las que Estados Unidos no se puede permitir perder su liderazgo en las tecnologías de desarrollo de software. A pesar de todo, no hacemos nada con nuestra fábrica de software que está en crisis. Puede llegar una época en la que sea menos costoso subcontratar el desarrollo Y el mantenimiento del software a terceras partes, situadas en cualquier otro lugar del mundo. Esas terceras partes estarán apoyadas por sus gobiernos, serán extremadamente competentes y, por lo tanto. muy competitivas. El software no es

metal, pero la competencia extranjera puede hacer que la situación de ambos llegue a ser muy semejante.

5.2. El software

Hace veinte años, menos del 1 por 100 de la gente podía describir de forma inteligente lo que significaba el "software de computadora". Hoy la mayoría de los profesionales y muchas personas en general creen que entienden el software. Pero, realmente lo entienden?

Una descripción del software de un libro de texto puede tener la siguiente forma: "Software: (1) instrucciones (programas de computadora) que cuando se ejecutan proporcionan la función Y el comportamiento deseado". (2) estructuras de datos que facilitan a los programas manipular adecuadamente la información, Y (3) documentos que describen la operación y el uso de los programas. No hay duda de que podrían ofrecerse otras definiciones más completas, Pero nosotros necesitamos algo más que una definición formal.

5.2.1. Consideraciones del software

Para poder comprender lo que es el software (y consecuentemente la ingeniería del software), es importante examinar las características del software que lo diferencian de otras cosas que los hombres pueden construir. Cuando se construye hardware, el proceso creativo humano (análisis, diseño, construcción, prueba) se traduce finalmente en una forma física. Si construimos una nueva computadora, nuestro boceto inicial, diagramas formales de diseño y prototipo de

prueba, evolucionan hacia un producto físico (tarjetas de circuitos impresos, fuentes de poder, etc.).

El software es un elemento del sistema que es lógico, en lugar de físico. Por tanto, el software tiene unas características considerablemente distintas a las del hardware:

" El software se desarrolla, no se fabrica en un sentido clásico."

Aunque existen algunas similitudes entre el desarrollo del software y la construcción del hardware, ambas actividades son fundamentalmente diferentes. En ambas actividades la buena calidad se adquiere mediante un buen diseño, pero la fase de construcción del hardware puede introducir problemas de calidad que no existen (o son fácilmente corregibles) en el software. Ambas actividades dependen de las personas, pero la relación entre la gente dedicada y el trabajo realizado es completamente diferente para el software. Ambas actividades requieren la construcción de un "producto", pero los métodos son diferentes.

Los costos del software se encuentran en la ingeniería. Esto significa que los proyectos de software no se pueden gestionar como si fueran proyectos de fabricación.

En la pasada década se ha tratado en la literatura el concepto de "fábrica de software". Es importante tener en cuenta que este término no implica que la Fabricación del hardware Y el desarrollo del software sean equivalentes. En vez

de ello, el concepto de fábrica de software recomienda el uso de herramientas para el desarrollo automático del software.

2. *El software no Falla.*

Para el hardware, la proporción de fallos como una función del tiempo. Esa relación. Denominada frecuentemente "curva de bañera". Indica que el hardware exhibe relativamente muchos fallos al principio de su vida (estos fallos son atribuibles normalmente a defectos del diseño o de la fabricación) una vez corregidos los defectos, la tasa de fallos cae hasta un nivel estacionario (bastante bajo, con un poco de optimismo) donde permanece durante un cierto período de tiempo. Sin embargo, conforme pasa el tiempo. Los fallos vuelven a presentarse a medida que los componentes del hardware sufren los efectos acumulativos de la suciedad, la vibración, los malos tratos, las temperaturas extremas y, muchos otros males externos. Sencillamente, el hardware comienza a Falla.

El software no es susceptible a los males del entorno que hacen que el hardware se dañe. Por tanto, en teoría, la curva de fallos para el software. Los defectos no detectados harán que falle el programa durante las primeras etapas de su vida. Sin embargo, una vez que se corrigen, suponiendo que no se introducen nuevos errores. la curva se aplana. es una gran simplificación de los modelos reales de fallos del software. Sin embargo, la implicación es clara el software no Falla, Pero se tiene que actualizar.

Esto, que parece una contradicción. Puede comprenderse mejor considerando. Durante su vida. el software sufre cambios (mantenimiento). Conforme se hacen los cambios, es bastante probable que se introduzcan nuevos defectos. Antes de que la curva pueda volver al estado estacionario original, se solicita otro cambio, haciendo que de nuevo se cree otro pico. Lentamente, el nivel mínimo de fallos comienza a crecer, el software se va deteriorando debido a los cambios.

Otro aspecto de ese deterioro ilustra la diferencia entre el hardware y el software. Cuando un componente de hardware se estropea, se sustituye por una "pieza de repuesto". No hay piezas de repuesto para el software. Cada fallo en el software indica un error en el diseño o en el proceso mediante el que se tradujo el diseño a código máquina ejecutable. Por tanto, el mantenimiento del software tiene una complejidad considerablemente mayor que la del mantenimiento del hardware.

3. La mayoría del software se construye a medida, en vez de ensamblar componentes existentes.

Consideremos la forma en la que se diseña se construye el hardware de control para un producto basado en microprocesador. El ingeniero de diseño construye un sencillo esquema de la circularía digital. hace algún análisis fundamental para asegurar que se realiza la función adecuada y va al catálogo de ventas de componentes digitales existentes. Cada circuito integrado (frecuentemente llamado un "LCI" o "pastillas') tiene un número de pieza, una función definida y, válida, una interfaz bien definida Y un conjunto estándar de criterios de

integración. Después de seleccionar cada componente. Puede solicitarse la compra.

Por desgracia. los diseñadores del software no disponen de esa comodidad que acabamos de describir. Con unas pocas excepciones, no existen catálogos de componentes de software. Se puede comprar software ya desarrollado. Pero sólo como una unidad completa, no como componentes que puedan reensamblarse en nuevos programas. Aunque se ha escrito mucho sobre "reusabilidad del software" (estamos comenzando a ver las primeras implementaciones con éxito de este concepto).

5.2.2. Componentes de software

El software de computadora es información que existe en dos formas básicas: componentes no ejecutables en la máquina y componentes ejecutables en la máquina. Por lo que a este capítulo respecta, sólo contemplaremos los componentes de software que conducen directamente a instrucciones máquina ejecutables. Todos los componentes de software tienen una *configuración,* de la que hablaremos adelante. Los componentes de software se crean mediante una serie de traducciones que hacen corresponder los requisitos del cliente con un código ejecutable en la máquina. Se traduce un modelo (prototipo) de requisitos a un diseño. Se traduce el diseño del software a una forma en un lenguaje que especifica las estructuras de datos, los atributos procedimentales Y los requisitos que atañen al software. La forma en lenguaje es procesada por un traductor que la convierte en instrucciones ejecutables en la máquina.

La reusabilidad es una característica importante para un componente de software de alta calidad . Es decir, el componente debe diseñarse e implementarse para que pueda volver a usarse en muchos programas diferentes. En los años sesenta se construyeron bibliotecas de subrutinas científicas reutilizables en una amplia serie de aplicaciones científicas y de ingeniería. Esas bibliotecas de subrutinas reutilizaban de forma efectiva algoritmos bien definidos. pero tenían un dominio de aplicación limitado. Hoy en día, Hemos extendido nuestra visión de reusabilidad para abarcar no sólo los algoritmos, sino también estructuras de datos. Un componente reusable de los años noventa en cápsula tanto datos como procesos en un único paquete (frecuentemente llamado *clase u objeto),* permitiendo al ingeniero de software crear nuevas aplicaciones a partir de trozos recusables. Por ejemplo, las interfaces interactivas de hoy en día se construyen frecuentemente a partir de componentes reusables que permiten la creación de ventanas gráficas, de menús emergentes y de una amplia variedad de mecanismos de interacción. Las estructuras de datos Y los detalles de procesados para construir la interfaz están contenidos en una biblioteca de componentes reusables orientados a la construcción de interfaces.

Los componentes de software se construyen mediante un lenguaje de programación que tiene un vocabulario limitado, una gramática definida explícitamente y reglas bien formadas de sintaxis y semántica. Estos atributos son esenciales para la traducción por la máquina. Las clases de lenguajes que se utilizan actualmente son lenguajes máquina, los lenguajes de alto nivel y los lenguajes no procedimentales.

Los lenguajes maquina son una representación simbólica del conjunto de instrucciones de *CPU*. Si un buen programador produce programas mantenibles y bien documentados, puede utilizar el lenguaje máquina para hacer un uso extremadamente eficiente de la memoria y para "optimizar" la velocidad de ejecución del programa. SI el programa está mal diseñado y tiene poca documentación, el lenguaje maquina puede convertirse en una pesadilla.

Los lenguajes de alto nivel permiten al programador y al programa idependizarse de la máquina Cuando se utiliza un traductor sofisticado, el vocabulario, la gramática la sintaxis y la semántica de un lenguaje de alto nivel pueden ser mucho más que los lenguajes máquina. De hecho, los compliadores e intérpretes de los lenguajes de alto nivel producen lenguaje máquina como salida.

Aunque hoy se utilizan cientos de lenguajes de programación, Poco más de una decena son lenguajes de programación de alto nivel con una gran aceptación en la industria. Después de casi treinta años desde su aparición, lenguajes como COBOL Y FORTAN siguen utilizándose mucho en la actualidad. Los lenguajes de programación modernos (lenguajes que soportan directamente las prácticas modernas para el diseño procedimental Y de datos) tales como Pascal, C y Ada se utilizan ampliamente. Los lenguajes orientados a los objetos como "C". pascal Y otros. están ganando cada vez más seguidores entusiastas.

Los lenguajes especializados (diseñados para ámbitos de aplicación específicos), como APL, LISP OPS5 y lenguajes descriptivos para redes neuronales artificiales.

Están teniendo cada vez mayor aceptación conforme las nuevas aplicaciones pasan de los laboratorios a la práctica.

El código máquina, los lenguajes ensambladores (nivel máquina) y los lenguajes de programación de alto nivel son normalmente considerados como las "tres primeras generaciones" de lenguajes de computadora. Con cualquiera de esos lenguajes, el programador ha de preocuparse tanto de la especificación de la estructura de la información como de la de control del propio programa. Por ello, los lenguajes de la tres, primeras generaciones se denominan *lenguajes procedimentales,*.

En la década pasada apareció un grupo de lenguajes *no procedimentales* de cuarta generación. En *vez* de requerir que quien desarrolla el software especifique los detalle, procedimentales, un programa en un lenguaje no procedimental es la "especificación del resultado deseado en *vez* de la especificación de la acción requerida para conseguir el resultado" . El software de soporte traduce la especificación del resultado en un programa máquina ejecutable.

Hasta la fecha, los lenguajes de cuarta generación se han utilizado en aplicaciones de base de datos y en otras áreas de procesamiento de datos para negocios. más adelante una exposición de su aplicación dentro del contexto de la ingeniería del software.

5.2.3. Aplicaciones del software

El software puede aplicarse en cualquier situación en la que se haya definido previamente un conjunto específico de pasos procedimertales (es decir, un algoritmo). (Excepciones notables a esta regla son el software de los sistemas expertos y de redes neuronales.) Para determinar la naturaleza de una aplicación de software, hay, dos factores importantes que se deben considerar: el contenido y el determinismo de la información. El *contenido* se refiere al significado y a la forma de la información de entrada Y de salida. Por ejemplo, muchas aplicaciones bancarias usan unos datos de entrada muy estructurados (una base de datos) y producen "Informes" con determinados formatos. El software que controla una máquina automática (un control numérico) actúa sobre elementos de datos discretos con una estructura limitada y produce órdenes concretas para la máquina en rápida sucesión.

El determinismo de la in información se refiere a la predecibilidad del orden y del tiempo de llegada de los datos. Un programa de ingeniería acepta datos que están en un orden predefinido, ejecuta el algoritmo sin interrupción y produce los datos resultantes en un informe o formato gráfico. Se dice que tales aplicaciones son determinadas. Un sistema operativo multiusuario. por otra parte, acepta entradas que tienen un contenido variado que se producen en instantes arbitrarios, ejecuta algoritmos que pueden ser interrumpidos por condiciones externas y produce una salida que depende de una función del entorno y del tiempo. Las aplicaciones con estas características se dice que son indeterminadas.

Algunas veces es difícil establecer categorías genéricas para las aplicaciones del software que sean significativas. Conforme aumenta la complejidad del software, es más difícil establecer comportamientos nítidamente separados. Las siguientes áreas del software indican la amplitud de las posibilidades de aplicación.

Software de sistemas. El software de sistemas es un conjunto de programas que han sido escritos para servir a otros programas, Algunos programas de sistemas (compiladores, editores utilidades de gestión de archivos) procesan estructuras de información complejas pero determinadas. Otras aplicaciones de sistemas (ciertos componentes del sistema operativo. Utilidades de manejo de periféricos. Procesadores de telecomunicaciones) procesan datos en gran medida indeterminados. En cualquier caso. el área del software de sistemas se caracteriza por una fuerte interacción con el hardware de la computadora: una gran utilización por múltiples usuarios; una operación concurrente que requiere una planificación, una comparación de recursos Y una sofisticada gestión de procesos: unas estructuras de datos complejas múltiples interfaces extremas.

Software de tiempo real. El software que mide/analiza/controla sucesos del mundo real conforme ocurren. se denomina de *tiempo real.* Entre los elementos del software de tiempo real se incluyen: un componente de adquisición de datos que recolecta y da formato a la información recibida del entorno externo, un componente de análisis que transforma la información según lo requiera la aplicación, un componente de control/salida que responda al entorno externo y un componente de monitorización que coordina todos los demás componentes, de

forma que pueda mantenerse la respuesta en tiempo real (típicamente en el rango de 1 milisegundo a 1 minuto). Hay que tener en cuenta que el término "tiempo real" tiene un significado diferente de "interactivo" o "tiempo compartido". Un sistema de tiempo real debe responder dentro de unas ligaduras estrictas de tiempo. El tiempo de respuesta de un sistema interactivo (o de tiempo compartido) puede ser normalmente sobrepasado sin que se produzca ningún desastre.

Software de gestión. El procesamiento de información comercial construye la mayor de las áreas de aplicación del software. Los "sistema discretos" (nóminas, cuentas de haberes/débitos, inventarios, etc.) han evolucionado hacia el software de sistemas de información de gestión (SIG), que accede a una o más bases de datos grandes que contienen información comercial. Las aplicaciones en este área reestructuran los datos existentes en orden a facilitar las operaciones comerciales o gestionar la toma de decisiones. Además de las tareas convencionales de procesamientos de datos, las aplicaciones de software de gestión también realizan cálculo interactivo (el procesamiento de transacciones en puntos de ventas).

Software de ingeniería y científico. El software de ingeniería Y científico está caracterizado por los algoritmos de "manejo de números". Las aplicaciones van desde la astronomía a la vulcanología, desde el análisis de la presión de los automotores a la dinámica orbital de las lanzadores espaciales y desde la biología molecular a la fabricación automática. Sin embargo, las nuevas aplicaciones del área de ingeniería/ciencia se han alejado de los algoritmos convencionales

numéricos. El diseño asistido por computadora (del inglés, CAD), la simulación de sistemas y otras aplicaciones interactivas, han comenzado a tomar características del software de tiempo real e incluso del software de sistemas.

Software preinstalado. Los productos inteligentes se han convertido en algo común en casi todos los mercados de consumo e industriales. El software preinstalado reside en memoria de sólo lectura y se utiliza para controlar productos sistemas de los mercados industriales y dé consumo. El software preinstalado puede ejecutar funciones muy, limitadas v curiosas (el control de las teclas de un horno de microondas) o suministrar una función significativa y con capacidad de control (funciones digitales en un automóvil, tales como control de la gasolina, indicaciones en el salpicadero, sistemas de frenado. etc.).

Software de computadoras personales. El mercado del software de las computadoras personales ha germinado en la pasada década. El procesamiento de textos. las hojas de cálculo. los gráficos por computadora, entretenimientos, gestión de bases de datos, aplicaciones financieras. de negocio, y personales. , redes o acceso a bases de datos externas son sólo algunas de los cientos de aplicaciones. De hecho, el software de las computadoras personales continúa representando uno de los diseños del software más innovadores en el campo del software.

Software de inteligencia artificial. El software de inteligencia artificial (IA) hace uso de algoritmos no numéricos para resolver problemas complejos para los que

no son adecuados el cálculo o el análisis directo. Actualmente, el área más activa de la IA es la de los *sistemas expertos*, también llamados *sistemas basados en el conocimiento*. Otras áreas de aplicación para el software de IA es el reconocimiento de patrones (Imágenes y voz), la prueba de teoremas y los juegos. En los últimos años ha surgido una nueva rama del software de IA llamada *redes neuronales artificiales* . Una red neuronal simula la estructura de proceso del cerebro (las funciones de la neurona biológica) y a la 91 larga puede llevar a una clase de software que pueda reconocer patrones complejos y aprender de "experiencia" pasada.

5.3. Problemática con el software

Muchos observadores de la industria han caracterizado los problemas asociados con el desarrollo del software como una "crisis". Sin embargo, lo que realmente tenemos puede ser algo bastante diferente.

La palabra "crisis", se define "un punto decisivo en el curso de algo; momento, etapa o evento decisivo o crucial". Sin embargo, para el software no ha habido ningún "punto decisivo", ningún "momento decisivo", solamente un lento cambio evolutivo. En la industria del software hemos tenido una "crisis" que ha estado con nosotros cerca de 30 años eso es una contradicción para el término.

Cualquiera que busque la palabra "crisis" en el diccionario encontrará otra definición: "el punto decisivo en el curso de una enfermedad, cuando se ve más claro si el paciente vivirá o morirá". Esta definición puede darnos una pista sobre la verdadera naturaleza de los problemas que han acosado al desarrollo del

software.Nosotros ya hemos alcanzado la etapa de crisis en el software de computadoras. Lo que realmente tenemos es una *aflicción crónica*. La palabra "aflicción" se define en el Webster como "algo que causa pena o desastre". Pero la clave de nuestro argumento es la definición del adjetivo "crónica": "muy duradero o que vuelve a aparecer con frecuencia. Continuando indefinidamente". Es bastante más preciso describir lo que hemos estado aguantando las tres ultimas décadas como una aflicción crónica que como una crisis. No hay curas milagrosas. Pero hay muchas formas con las que podemos reducir la pena a medida que nos esforcemos por descubrir un remedio.

Tanto si lo llamamos crisis del software como mal del software, el término alude a un conjunto de problemas que aparecen en el desarrollo del software de computadoras. Los problemas no se limitan al software que "no funciona correctamente". Es más, el mal abarca los problemas asociados a cómo desarrollar software. Cómo mantener el volumen cada vez mayor de software existente y cómo poder esperar mantenernos al corriente de la demanda creciente de software. Aunque se puede criticar la referencia a crisis o incluso a aflicción por ser melodramática, las frases resultan útiles por referirse a verdaderos problemas que se encuentran en todas las áreas del desarrollo del software.

5.3.1. Dificultades

Los problemas que afligen al desarrollo del software se pueden caracterizar bajo muchas perspectivas diferentes, pero los responsables de los desarrollos de software se centran sobre los aspectos de "fondo": (1) la planificación y estimación

de costos son frecuentemente muy imprecisas; (2) la "productividad" de la comunidad del software no se corresponde con la demanda de sus servicios y (3) la calidad del software no llega a ser a veces ni aceptable. Se han experimentado desajustes en los costos de hasta un orden de magnitud. Se ha errado en la planificación en meses o años. Se ha hecho muy poco para mejorar la productividad de los trabajadores del software. Los errores en los nuevos programas producen en los clientes insatisfacción y falta de confianza. Tales problemas son sólo las manifestaciones más ,Visibles de otras dificultades del software:

* No tenemos tiempo de recoger datos sobre el proceso de desarrollo del software. Sin datos históricos como guía, la estimación no ha sido buena Y los resultados previstos muy pobres. Sin una indicación sólida de la productividad, no podemos evaluar con precisión la eficacia de las nuevas herramientas, técnicas o estándares.
* La insatisfacción del cliente con el sistema "terminado" se produce demasiado frecuentemente. Los proyectos de desarrollo del software se acometen frecuentemente con sólo una vaga indicación de los requisitos del cliente. Normalmente, la comunicación entre el cliente y el que desarrolla el software es muy escasa.
* La calidad del software es normalmente cuestionable. Hemos empezado a comprender recientemente la importancia de la prueba sistemática y técnicamente completa del software. Están comenzando a emerger conceptos cuantitativos sólidos sobre la fiabilidad del software y las garantías de calidad

- El software existente puede ser muy difícil de mantener. La tarea de mantenimiento del software se lleva la mayor parte de todo el dinero invertido en el software. El mantenimiento no se ha considerado un criterio importante en la aceptación del software.

Hemos presentado primero las malas noticias. Ahora las buenas: todos los problemas descritos anteriormente pueden corregirse. La clave está en dar un enfoque de ingeniería al desarrollo del software, 'unto con la mejora continua de las técnicas Y de las herramientas.

Sin embargó, seguirá existiendo un problema (podríamos decir que son cosas de la vida). El software absorberá cada vez mayores porcentajes del costo de desarrollo global de los sistemas basados en computadora. En Estados Unidos se gasta cerca de 200 000 millones de dólares cada año en el desarrollo, compra y mantenimiento de software de computadora. Nos hemos tomado más en serio los problemas asociados con el desarrollo del software.

5.3.2. Causalidad

Los problemas asociados con la crisis del software se han producido por el propio carácter del software y por los errores de las personas encargadas del desarrollo del mismo. Sin embargo, es posible que esperemos mucho en poco tiempo. Después de todo, nuestra experiencia no va más allá de cuarenta años aproximadamente.

Hemos tratado brevemente el carácter del software de computadora con anterioridad Recordemos: el software es un elemento lógico en vez de físico; por

tanto. el éxito se mide por la calidad de una única entidad en vez de por muchas entidades fabricadas. El software no se rompe. Si se encuentran fallos, lo más probable es que se introdujeran inadvertidamente durante el desarrollo y, no se detectaran durante la prueba. Reemplazamos las "partes defectuosas" durante el mantenimiento del software, pero tenemos muy pocas piezas de repuesto, incluso ninguna; es decir, el mantenimiento incluye normalmente la corrección o modificación del diseño.

La naturaleza lógica del software representa un desafío para la gente que lo desarrolla. Por primera vez hemos aceptado el reto de comunicarnos con un alienígena inteligente una máquina. El desafío intelectual del desarrollo de software es seguramente una de las causas de la crisis del software, pero los problemas tratados anteriormente han sido causados por defectos humanos.

Frecuentemente, los responsables del desarrollo del software han sido ejecutivos de nivel medio alto. sin conocimientos de software. Un antiguo axioma de desarrollo dice: "Un buen desarrollador puede desarrollar cualquier proyecto'". Nosotros debemos añadir: "...Si está dispuesto a aprender las nuevas técnicas que pueden utilizarse para el desarrollo del proyecto. a aplicar métodos efectivos de control. a ignorar la mitología y a llegar a conocer una tecnología que cambia ' rápidamente'. El desarrollador debe comunicarse con todos los componentes implicados en el desarrollo del software clientes, realizadores del software, equipo de soporte y otros. La comunicación puede romperse debido a que se comprenden mal las características especiales del software, los problemas

particulares asociados con su desarrollo. Cuando esto ocurre. los problemas asociados con la crisis del software se multiplican.

Los trabajadores del software (en la pasada generación se llamaban *programadores;* en esta generación se ganará el título de *ingenieros de software)* han tenido muy poco entrenamiento formal en las nuevas técnicas de desarrollo de software. En muchas organizaciones reina una suave forma de anarquía. Cada individuo enfoca su tarea de 'escribir programas" con la experiencia obtenida en trabajos anteriores. Algunas personas desarrollan un método ordenado y eficiente de desarrollo del software mediante prueba y error, pero muchos otros desarrollan malos hábitos que dan como resultado una pobre calidad y mantenibilidad del software.

Todos nos resistimos al cambio. Sin embargo, es verdaderamente irónico que, mientras el potencial de cálculo (hardware) experimenta enormes cambios. la gente del software, responsable de aprovechar dicho potencial, se oponga normalmente a los cambios cuando se discuten y se resista al cambio cuando se introduce. Puede que ésta sea la causa real de la crisis del software.

5.4. Mitos del Software

Muchas de las causas de la crisis del software se pueden encontrar en una mitología que surge durante los primeros años del desarrollo del software. A diferencia de los mitos antiguos, que ofrecían a los hombres lecciones dignas de

tener en cuenta, los mitos del software propagaron información errónea y confusión. Los mitos del software tienen atributos que los hacen insidiosos; por ejemplo. aparecieron como declaraciones razonables de hechos (algunas veces conteniendo elementos verdaderos). Tuvieron un sentido intuitivo y frecuentemente fueron promulgados por expertos que "estaban al día".

Hoy, la mayoría de los profesionales competentes consideran a los mitos por lo que son actitudes erróneas que han causado serios problemas, tanto a los desarrolladores como a los técnicos. Sin embargo, las ,viejas actitudes y hábitos son difíciles de modificar Y, cuando vamos hacia la quinta década del software, todavía se cree en algunos de los mitos del software.

Mitos de gestión. Los desarrolladores con responsabilidad sobre el software. Como los desarrolladores en la mayoría de las disciplinas, están normalmente bajo la presión de cumplir los presupuestos. Hacer que no se retrase el proyecto y mejorar la calidad. Igual que se agarra al vacío una persona que se ahoga, un gestor de software se agarra frecuentemente a un mito del software, aunque tal creencia sólo disminuya la presión temporalmente.

Mito. Existen manuales que están llenos de estándares y procedimientos para construir software.

No les proporciona ya a mi gente todo lo que necesita saber?

Realidad. Está bien que un manual exista, pero se usa?

Conocen los trabajadores su existencia Refleja las prácticas modernas de desarrollo de software?

Es completo En muchos casos. La respuesta a todas estas preguntas es " no .

Mito. Nuestra gente dispone de las herramientas de desarrollo de software más avanzadas; después de todo, les compramos computadoras nuevas.

Realidad. Se necesita mucho más que el último modelo de computadora grande (o de PC) para hacer desarrollo de software de gran calidad. Las herramientas de ingeniería del software asistida por computadora (CASE), aunque la mayoría todavía no se usen, son más importantes que el hardware para conseguir buena calidad Y productividad.

Mito. Si fallamos en la planificación, podemos añadir más programadores y adelantar el tiempo perdido.

Realidad. El desarrollo de software no es un proceso mecánico como la fabricación. En palabras de añadir gente a un proyecto de software retrasado lo retrasa aún más". Al principio, esta declaración puede parecer un contrasentido. Sin embargo, cuando se añaden nuevas personas, la necesidad de aprender y comunicarse con el equipo puede y hace que se reduzca la cantidad de tiempo gastado en el desarrollo productivo. Puede añadirse gente, pero sólo de una manera planificada y bien coordinada.

Mitos del cliente. Un cliente que solicita una aplicación de software puede ser una persona del despacho de al lado, un grupo técnico de la sala de abajo. el departamento de ventas o una compañía exterior que solicita un software bajo contrato. En muchos casos, el cliente cree en los mitos que existen sobre el software. Debido a que los gestores y trabajadores responsables hacen muy poco para corregir la mala información. Los mitos conducen a que el cliente se cree

una falsa expectativa y finalmente, quede insatisfecho con el que desarrolla el software.

Mito. Una declaración general de los objetivos es suficiente para comenzar a escribir los programas podemos dar los detalles más adelante.

Realidad. Una mala definición inicial es la principal causa del trabajo baldío en software. Es esencial una descripción formal Y detallada del ámbito de la información. funciones. rendimiento, interfaces, ligaduras del diseño y criterios de validación. Estas características pueden determinarse sólo después de una exhaustiva comunicación entre el cliente y el analista.

Mito. Los requisitos del proyecto cambian continuamente, pero los cambios pueden acomodarse fácilmente. va que el software es flexible.

Realidad. Es verdad que los requisitos del software cambian, pero el impacto del cambio varía según el momento en que se introduzca. Si se pone cuidado al dar la definición inicial, los cambios solicitados al principio pueden acomodarse fácilmente. El cliente puede revisar los requisitos y, recomendar las modificaciones con relativamente poco impacto en el costo. Cuando los cambios se solicitan durante el diseño del software, el impacto en el costo crece rápidamente. Ya se han acordado los recursos a utilizar?

se ha establecido un esqueleto del diseño. Los cambios pueden producir trastornos que requieran recursos adicionales e importantes modificaciones del diseño; es decir, costo adicional. Los cambios en la función, rendimiento, interfaces u otras características. durante la implementación (codificación y prueba) pueden tener un impacto importante sobre el costo. Cuando se solicitan

al final de un proyecto, los cambios pueden producir un orden de magnitud más caro que el mismo cambio pedido al principio.

Mitos de los desarrolladores. Los mitos en los que aún creen muchos desarrolladores se han ido fomentando durante cuatro décadas de cultura informática. Como ya indicamos anteriormente en este capítulo, durante los primeros días del desarrollo de software. la programación se veía como un arte. Las viejas formas y actitudes tardan en morir.

Mito. Una vez que escribimos el programa y hacemos que funcione, nuestro trabajo ha terminado.

Realidad. Alguien dijo una vez: "Cuanto más pronto se comience a escribir código, más se tardará en terminarlo". Los datos industriales indican que entre el 50 v el 70 por 100 de todo el esfuerzo dedicado a un programa se realizará después de que se le haya entregado al cliente por primera vez.

Mito. Hasta que no tengo el programa "ejecutándose", realmente no tengo forma de comprobar su calidad.

Realidad. Desde el principio del proyecto se puede aplicar uno de los mecanismos más efectivos para garantizar la calidad del software la *revisón técnica formal.* La revisión del software es un "filtro de calidad" que se ha comprobado que es más efectivo que la prueba, para encontrar ciertas clases de defectos en el software.

Mito. Lo único que se entrega al terminar el proyecto es el programa funcionando.

Realidad. Un programa que funciona es sólo una parte de una *configuración del software* que incluye todos los elementos de La documentación es la base de un buen desarrollo y, lo que es más importante, proporciona guías para la tarea de mantenimiento del software.

Muchos profesionales del software reconocen la falacia de los mitos descritos anteriormente. Lamentablemente, las actitudes y métodos habituales fomentan una pobre gestión y malas prácticas técnicas, incluso cuando la realidad dicta un método mejor. El reconocimiento de las realidades del software es el primer paso hacia la formulación de soluciones prácticas para el desarrollo del mismo.

5.5. Paradigmas del Sofware

El mal que ha infectado el desarrollo del software no va a desaparecer de la noche a la mañana. Reconocer los problemas y sus causas y demoler los mitos del software son los primeros pasos hacia las soluciones. Pero las propias soluciones tienen que proporcionar asistencia práctica a la persona que desarrolla software., mejorar la calidad del software y, por último, permitir al mundo del software mantenerse en paz con el mundo del hardware.

No existe un único enfoque mejor para solucionar el mal del software. Sin embargo, mediante la combinación de métodos completos para todas las fases del desarrollo del software., mejores herramientas para automatizar estos métodos, bloques de construcción más potentes para la implementación del software, mejores técnicas para la garantía de calidad del software y una filosofía

predominante para la coordinación, control y gestión, podemos conseguir una disciplina para el desarrollo del software una disciplina llamada *Ingeniería del software.*

5.5.1. Definición de la ingeniería de Software

UNA DEFINICIÓN

Una de las primeras definiciones de ingeniería del software fue la propuesta por

Fritz Bauer

El establecimiento, uso de principios de ingeniería robustos. Orientados a obtener, software económico que sea fiable y funcione de manera eficiente sobre máquinas reales.

Aunque se han propuesto muchas más definiciones generales. todas refuerzan la importancia de una disciplina de ingeniería para el desarrollo del software.

La ingeniería del software surge de la ingeniería de sistemas y de hardware. Abarca un conjunto de tres elementos clave métodos, herramientas y procedimientos que facilitan al desarrollador controlar el proceso del desarrollo del software y suministrar a los que practiquen dicha ingeniería las bases para construir software de alta calidad de una forma productiva.

Los *métodos* de la ingeniería del software indican "cómo" construir técnicamente el software. Los métodos abarcan un amplio espectro de tareas que incluyen:

planificación y estimación de proyectos, análisis de los requisitos del tema Y del software. diseño de estructuras de datos, arquitectura de programas y procedimientos algoritmos, codificación, prueba Y mantenimiento. Los métodos

de la ingeniería del software introducen frecuentemente una notación especial orientada a un lenguaje o gráfica y un conjunto de criterios para la calidad del software.

Las herramientas de la ingeniería del software suministran un soporte automático o semiautomático para los métodos. Hoy existen herramientas para soportar cada uno de los métodos mencionados anteriormente. Cuando se integran las herramientas de forma que la información creada por una herramienta pueda ser usada por otra, se establece un sistema para el soporte del desarrollo del software, llamado *ingeniería del software asistida por computadora* (del inglés, CASE). CASE combina software, hardware Y bases de datos sobre ingeniería del software (una estructura de datos que contenga la información relevante sobre el análisis, diseño, codificación y prueba) para crear un entorno de ingeniería del software el análogo al diseño/Ingeniería asistido por computadora, CAD/CAE (de las siglas en inglés) para el hardware.

Los procedimientos de la ingeniería del software son el pegamento que junta los métodos y las herramientas y facilita un desarrollo racional y oportuno del software de computadora. Los procedimientos definen la secuencia en la que se aplican los métodos, las entregas (documentos, informes, formas, etc.) que se requieren, los controles que ayudan a asegurar la calidad y coordinar los cambios, y las directrices que ayudan a los desarrolladores del software a evaluar el progreso.

La ingeniería del software está compuesta por una serie de pasos que abarcan los métodos, las herramientas y los procedimientos antes mencionados. Estos pasos se denominan frecuentemente *paradigmas de la ingeniería del software.* La elección de un paradigma para la ingeniería del software se lleva a cabo de acuerdo con la naturaleza del proyecto y de la aplicación.

6.5.2. El ciclo de vida clásico

Algunas veces llamado "modelo en cascada", el paradigma del ciclo de vida exige un enfoque sistemático y secuencias del desarrollo del software que comienza en el nivel del sistema Y progresa a través del análisis, diseño, codificación. prueba y mantenimiento. Modelizado a partir del ciclo convencional de una ingeniería, el paradigma del ciclo de vida abarca las siguientes actividades:

Ingeniería Y análisis del sistema. Debido a que el software es siempre parte de un sistema mayor. el trabajo comienza estableciendo los requisitos de todos los elementos del sistema Y luego asignando algún subconjunto de estos requisitos al software. Este planteamiento del sistema es esencial cuando el software debe interrelacionarse con otros elementos, tales como hardware, personas y bases de datos. La ingeniería y el análisis del sistema abarca los requisitos globales a nivel del sistema con una pequeña cantidad de análisis y de diseño a un nivel superior.

Análisis de los requisito del software. El proceso de recopilación de los requisitos se centra e intensifica especialmente para el software. Para comprender la naturaleza de los programas que hay que construir,. el ingeniero de software ("analista") debe comprender el ámbito de la información del software así como la

función, el rendimiento y las interfaces requeridas. Los requisitos, tanto del sistema como del software, se documentan y se revisan con el cliente.

Diseño. El diseño del software es realmente un proceso múltiple que se enfoca sobre cuatro tributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interface. El proceso de diseño traduce los requisitos en una representación del software que pueda ser establecida de forma que obtenga la calidad requerida antes de que comience la codificación. Al igual que los requisitos, el diseño se documenta y forma parte de la configuración del software.

Codificación. El diseño debe traducirse en una forma legible para la máquina. El paso de codificación realiza esta tarea. Si el diseño se realiza de una manera detallada, la codificación puede realizarse mecánicamente.

Prueba. Una vez que se ha generado el código, comienza la prueba del programa. La prueba se entra en la lógica interna del software. Asegurando que todas las sentencias se han probado. Y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.

Mantenimiento. El software, indudablemente, sufrirá cambios después de que se entregue al cliente (una posible excepción es el software preinstalado). Los cambios ocurrirán debido a que se hayan encontrado errores, a que el software deba adaptarse a cambios del entorno externo (un cambio solicitado debido a que se tiene un nuevo sistema operativo o dispositivo periférico), o debido a que el

cliente requiera ampliaciones funcionales o del rendimiento. El mantenimiento del software aplica cada uno de los pasos precedentes del ciclo de vida a un programa existente en vez de a uno nuevo.

El ciclo de vida clásico es el paradigma más antiguo y más ampliamente usado en la ingeniería del software. Sin embargo, con el paso de unos cuantos años, se han producido críticas al paradigma, incluso por seguidores activos. que cuestionan su aplicabilidad a todas las situaciones. Entre los problemas que se Presentan algunas veces, cuando se aplica el paradigma del ciclo de la vida clásico. se encuentran:

1 Los proyectos reales raramente siguen el flujo secuencias que propone el modelo. Siempre hay iteraciones y se crean problemas en la aplicación del paradigma.

2. Normalmente, es difícil para el cliente establecer explícitamente al principio todos los requisitos. El ciclo de vida clásico lo requiere y tiene dificultades en acomodar posibles incertidumbres que pueden existir al comienzo de muchos proyectos.

3. El cliente debe tener paciencia. Hasta llegar a las etapas finales de] desarrollo del proyecto, no estará disponible una versión operativa del programa. Un error importante no detectado hasta que el programa esté funcionando puede ser desastroso.

Cada uno de estos problemas es real. Sin embargo, el paradigma clásico del ciclo de vida tiene un lugar definido e importante dentro del trabajo realizado en ingeniería del software. Suministra una plantilla en la que pueden colocarse los métodos para el análisis, diseño, codificación, prueba y mantenimiento. Además, veremos que los pasos del paradigma clásico del ciclo de vida son muy similares a los pasos genéricos aplicables a todos los paradigmas del software.

El ciclo de vida clásico sigue siendo el modelo procedimental más ampliamente usado por los ingenieros del software. A pesar de sus inconvenientes, es significativamente mejor que desarrollar el software sin guías.

5.5.3. Construcción de prototipos

Normalmente un cliente define un conjunto de objetivos generales para el software. pero no identifica los requisitos detallados de entrada, proceso o salida. En otros casos, el programador puede no estar seguro de la eficiencia de un algoritmo. De la adaptabilidad de un sistema operativo o de la forma en que debe realizarse la interacción hombre-máquina. En estas y muchas otras situaciones, puede ser mejor método de ingeniería del software la construcción de un prototipo.

La construcción de prototipos es un proceso que facilita al programador la creación de un modelo del software a construir. El modelo tomará una de las tres formas siguientes: (1) un prototipo en papel o un modelo basado en PC que describa la interacción hombre-máquina, de forma que facilite al usuario la describa la interacción de cómo se producirá tal interacción. (2) un prototipo que

implemente algunos subconjuntos de la función requerida del programa deseado, o (3) un programa existente que ejecute parte o toda la función deseada. pero que tenga otras características que deban ser mejoradas en el nuevo trabajo de desarrollo.

Como en todos los métodos de desarrollo de software, la construcción de prototipos comienza con la recolección de los requisitos. El técnico Y el cliente se reúnen Y definen los objetivos globales para el software, identifican todos los requisitos conocidos Y perfilan las áreas en donde será necesario una mayor definición. Luego se produce un "diseño rápido". El diseño rápido se enfoca sobre la representación de los aspectos del software visibles al usuario (métodos de entrada, formatos de salida). El diseño rápido conduce a la construcción de un prototipo.

El prototipo es evaluado por el cliente/usuario y se utiliza para refinar los requisitos del software a desarrollar. Se produce un proceso interactivo en el que el prototipo "este a tiempo" para que satisfaga las necesidades del cliente, al mismo tiempo que facilita al que lo desarrolla una mejor comprensión de lo que hay que hacer.

Idealmente, el prototipo sirve como mecanismo para identificar los requisitos del software. Si se a construir un prototipo que funcione. el realizador intenta hacer uso de fragmentos de programas existentes o aplica herramientas que faciliten la rápida generación de programas que funcionen.

Pero, ¿qué debemos hacer con el prototipo cuando ya ha servido para el propósito establecido?

En la. Mayoría de los proyectos. El primer sistema construido apenas es utilizable. Puede ser demasiado lento. Difícil de usar o las dos cosas. No hay más alternativa que comenzar de nuevo y Construir una versión rediseñada que resuelva los problemas que se presenten Cuando se utiliza nuevo concepto de sistema o de tecnología. Hay que construir un sistema para desecharlo, porque incluso la mejor, planificación no puede asegurar que vaya a ser bueno la primera vez. Por tanto, la cuestión no es si hay que construir un sistema piloto, tirarlo. Se tirará. La única cuestión es si planificar de antemano la construcción de algo que se va a desechar, o prometer entregar el desecho a los clientes.

El prototipo puede servir como "primer sistema" Pero esto puede ser una visión idealizada. Al igual que en el ciclo de vida clásico, la construcción de prototipos como paradigma para la ingeniería del software, puede ser problemática por las siguientes razones:

1 . El cliente ve funcionando lo que parece ser una primera versión del software, ignorando que el prototipo se ha hecho con "plastilina Y alambres", ignorando que, por las prisas en hacer que funcione, no hemos considerado los aspectos de calidad o de mantenimiento del software a largo plazo.

Cuando se le informa de que el producto debe ser reconstruido, el cliente se vuelve loco y solicita que se apliquen "cuantas mejoras" sean necesarias

para hacer del prototipo un producto final que funcione. El gestor del desarrollo del software cede demasiado a menudo.

2. El técnico de desarrollo, frecuentemente. impone ciertos compromisos de implementación con el fin de obtener un prototipo que funcione rápidamente. Puede que utilice un sistema operativo o un lenguaje de programación inapropiados, simplemente porque va está disponible y es conocido, puede que implemente ineficientemente un algoritmo, sencillamente para demostrar su capacidad. Después de algún tiempo, el técnico puede haberse familiarizado con esas elecciones y haber olvidado las razones por las que eran inapropiadas. La elección menos ideal forma ahora parte integral del sistema.

Aunque pueden aparecer problemas. la construcción de prototipos es un paradigma efectivo para la ingeniería del software. La clave está en definir al comienzo las reglas del juego. esto es, el cliente v el técnico deben estar de acuerdo en que el prototipo se construya para servir sólo como un mecanismo de definición de los requisitos. Posteriormente. ha de ser descartado (al menos en parte) y debe construirse el software real. con los ojos puestos en la calidad en el mantenimiento.

5.5.4. Modelo de Espiral

El modelo en espiral para la ingeniería del software ha sido desarrollado para cubrir las mejores características tanto del ciclo de vida clásico. como de la

creación de prototipos. añadiendo al mismo tiempo un nuevo elemento: el análisis de riesgo. que falta en esos paradigmas. El modelo., representado mediante la espiral define cuatro actividades principales. representadas por los cuatro cuadrantes.

1. *Planificación*.- determinación de objetivos, alternativas y restricciones
3. *Análisis de riesgo*.- análisis de alternativas e identificación/resolución de riesgos
3. *Ingeniería*.- desarrollo del producto de "siguiente nivel"
4. *Evaluación del cliente*.- valoración de los resultados de la ingeniería

Un aspecto intrigante del modelo en espiral se hace evidente cuando consideramos la dimensión radial Con cada iteración alrededor de la espiral (comenzando en el centro Y siguiendo hacia el exterior), se construyen sucesivas versiones del software, cada vez más completas. Durante la primera vuelta alrededor de la espiral se definen los objetivos, las alternativas y las restricciones. y se analizan e identifican los riesgos.

Si el análisis de riesgo indica que hay una incertidumbre en los requisitos. se puede usar la creación de prototipos en el cuadrante de ingeniería para dar asistencia tanto al encargado del desarrollo como al cliente. Se pueden usar simulaciones y otros modelos para definir más el problema refinar los requisitos.
El cliente evalúa el trabajo de ingeniería (cuadrante de evaluación del cliente) y sugiere modificaciones. En base a los comentarios del cliente se produce la

siguiente fase de planificación y de análisis de riesgo. En cada ciclo alrededor de la espiral, la culminación del análisis de riesgo resulta en una decisión de "seguir o no seguir". Si los riesgos son demasiado grandes, se puede dar por terminado el proyecto.

Sin embargo, en la mayoría de los casos. se sigue avanzando alrededor del camino de la espiral. y ese camino lleva a los desarrolladores hacia fuera, hacia un modelo más completo del sistema, y, al final, al propio sistema operacional.

Cada vuelta alrededor de la espiral requiere ingeniería (cuadrante inferior derecho). que se puede llevar a cabo mediante el enfoque del ciclo de vida clásico o de la creación de prototipos. Debe tenerse en cuenta que el número de actividades de desarrollo que ocurren en el cuadrante inferior derecho aumenta al asearse del centro de la espiral.

El paradigma del modelo en espiral para la ingeniería del software es actualmente el enfoque más realista para el desarrollo de software y de sistemas a gran escala. Utiliza un enfoque "evolutivo" para la ingeniería del software, permitiendo al desarrollador y al cliente entender y reaccionar a os riesgos en cada nivel evolutivo. Utiliza la creación de prototipos como un mecanismo de reducción del riesgo, pero, lo que es mas importante, permite a quien desarrolla aplicar el enfoque de creación de prototipos en cualquier etapa de la evolución del producto. Mantiene el enfoque sistemático correspondiente a los pasos sugeridos por el ciclo de vida clásico. pero incorporándola dentro de un marco de trabajo

interactivo que refleja de forma más realista el mundo real. El modelo en espiral demanda una consideración directa de riesgos técnicos en todas las etapas del proyecto y, si se aplica adecuadamente, debe reducir los riesgos antes de que se conviertan en problemáticos.

Pero, al igual que otros paradigmas, el modelo en espiral no es la panacea. Puede ser difícil convencer a grandes clientes (particularmente en situaciones bajo contrato) de que el enfoque evolutivo es controlable. Requiere una considerable habilidad para la valoración del riesgo, y cuenta con esta habilidad para el éxito. Si no se descubre un riesgo importante, indudablemente surgirán problemas. Por último. el modelo en sí mismo es relativamente nuevo y no se ha usado tanto como el cielo de ,lda o la creación de prototipos. Pasarán unos cuantos años antes de que se pueda determinar con absoluta certeza la eficacia de este importante nuevo paradigma.

5.5.5 Técnicas de cuarta generación

El término "técnicas de cuarta generación" (T4G) abarca un amplio espectro de herramientas de software que tienen algo en común: todas facilitan. al que desarrolla el software. la especificación de algunas características del software a alto nivel. Luego. la herramienta genera automáticamente el código fuente basándose en la especificación del técnico. Cada vez parece más evidente que cuanto mayor sea el nivel en el que se especifique el software. Más rápido se podrá construir el programa. El paradigma T4G para la ingeniería del software se

orienta hacia la posibilidad de especificar el software a un nivel más próximo al lenguaje natural o en una notación que proporcione funciones significativas.

Actualmente, un entorno para el desarrollo de software que soporte el paradigma T4G puede incluir todas o algunas de las siguientes herramientas: lenguajes no procedimentales para consulta a bases de datos. generación de informes, manipulación de datos, interacción y definición de pantallas, generación de código. facilidades gráficas de alto nivel Y facilidades de hoja de cálculo. Todas estas herramientas están disponibles, pero sólo para ámbitos de aplicación muy, específicos. Actualmente. no existe un entorno T4G que pueda aplicarse con igual facilidad a todas las categorías de aplicaciones de, software descritas anteriormente.

Al igual que otros paradigmas. T4G comienza con el paso de recolección de requisitos. Idealmente, el cliente describe los requisitos, que son, a continuación, traducidos directamente a un prototipo operativo. Sin embargo., en la práctica no se puede hacer eso. El cliente puede no estar seguro de lo que necesita, puede ser ambiguo en la especificación de hechos que le son conocidos Y puede no desear o ser incapaz de especificar la información en la forma en que una herramienta T4G puede aceptarla. Además, las herramientas actuales de T4G no son lo suficientemente sofisticadas como para entender el lenguaje "natural", y no lo serán por algún tiempo. En este momento el diálogo cliente-técnico descrito por los otros paradigmas sigue siendo una parte esencial del enfoque T4G.

Para aplicaciones pequeñas. se puede ir directamente desde el paso de recolección de requisitos al paso de implementación, usando un *lenguaje de cuarta generación* no procedimental . Sin embargo. es necesario un mayor esfuerzo para desarrollar una estrategia de diseño para el sistema. Incluso si se utiliza un T4G . El uso de T4G sin diseño (para grandes proyectos) causará las mismas dificultades (poca calidad. Mantenimiento pobre, mala aceptación por el cliente) que se encuentran cuando se desarrolla software mediante los enfoques convencionales.

La implementación mediante un T4G permite, al que desarrolla el software. Centrarse en la representación de los resultados deseados. que es lo que se traduce automáticamente en un código fuente que produce dichos resultados. obviamente. debe existir una estructura de datos con información relevante a la que el T4G pueda acceder rápidamente. Mas adelante daremos una descripción más detallada de los lenguajes de cuarta generación.

Para transformar una implementación T4G en un producto. el que lo desarrolla debe dirigir una prueba completa. Desarrollar una documentación con sentido y ejecutar el resto de actividades de "transición" requeridas en los otros Paradigmas de ingeniería del software. Además. el software desarrollado con T4G deber ser construido de forma que facilite la realización del mantenimiento de forma expeditiva. Ha habido mucha controversia y un considerable debate sobre el uso de] paradigma T4G. Los defensores aducen reducciones drásticas en el tiempo de desarrollo del software y una mejora significativa en la productividad de la

gente que construye el software. Los detractores aducen que las herramientas actuales de T4G no son más fáciles de utilizar que los lenguajes de programación, que el código fuente producido por tales herramientas es "ineficiente" y que el mantenimiento de grandes sistemas de software desarrollados mediante T4G. es cuestionable. Ambas posturas tienen su parte de razón. Aunque es difícil separar los hechos de las suposiciones (existen pocos estudios controlados hasta el momento), es posible resumir el estado actual de los métodos de T4G:

1. Con muy pocas excepciones, el ámbito de aplicación actual de las T4G está limitado a las aplicaciones de sistemas de información de gestión, concretamente al análisis de información v la obtención de informes relativos a grandes bases de datos. Sin embargo, las nuevas herramientas CASE soportan ahora el uso de T4G para la generación automática de "esquemas de código" para aplicaciones de ingeniería Y de tiempo real.
2. Los datos preliminares recogidos en compañías que usan T4G parecen indicar que el tiempo requerido para producir software se reduce mucho para aplicaciones pequeñas y de tamaño medio, y que la cantidad de análisis y diseño para las aplicaciones pequeñas, también se reduce.
3. Sin embargo, el uso de T4G para grandes trabajos de desarrollo de software exige el mismo o más tiempo de análisis, diseño y prueba (actividades de ingeniería del software). Perdiéndose así un tiempo sustancial que se ahorra mediante la eliminación de la codificación.

Resumiendo, las técnicas de cuarta generación ya se han convertido en una parte importante del desarrollo de software en el área de aplicaciones de sistemas de información y probablemente llegarán a usarse bastante en aplicaciones de ingeniería y de tiempo real durante la segunda mitad de la década de los noventa. La demanda de software continuará creciendo durante el resto de este siglo, pero los métodos Y los paradigmas convencionales contribuirán probablemente cada vez menos al desarrollo del mismo. Las técnicas de cuarta generación llenarán el hueco existente.

5.6.- Selección del Software

Muchos de los aspectos de interés señalados en la elección de hardware también se aplican al software. La determinación de cuál software comercial es el correcto para una tarea particular y el acuerdo de los términos contractuales son responsabilidad de la organización.

Una de la tareas más difíciles en la elección del software, una vez que se conocen los requerimientos del sistema, es el determina, si un cierto paquete de software cumple con los requerimientos. Después de la selección inicial, es necesario escudriñar un poco mas para determinar lo deseable de un software particular comparado con otros candidatos.

Algoritmo de programación y compilación

Eficiencia de código

Algoritmo de manejo de la memoria virtual

Eficiencia en el manejo de archivos

Manejo de interrupciones

Método de indexación

manejo de almacenamientos temporales múltiples

Procedimientos de procesamiento en la comunicación

Las comparaciones se hacen puramente en una base cuantitativa. No relacionan tiempo necesario de aprendizaje para acostumbrarse al sistema o a la calidad del software del sistema (tal como la calidad del diagnóstico producido durante la compilación o la eficiencia del código objeto producido).

Las pruebas nos proporcionan una seguridad razonable de que los programas que en el momento se usan en un sistema existente se puedan convertir al nuevo sistema o que la nueva máquina los procesará eficientemente, incluso cuando estén convertidos. Los proveedores también podrán hacer afirmaciones, en el sentido de que un sistema específico es capaz de realizar tareas adicionales que otro sistema no puede.

5.6.1. - FLEXIBILIDAD

Debe incluir la capacidad de cumplir con los requerimientos cambiantes y las diferentes necesidades del usuario. Aunque no es deseable la flexibilidad excesiva. Varía de acuerdo con los tipos de hardware con los que trabajará. La flexibilidad en la salida es igualmente importante.

5.6.2. - PREVISIONES DE AUDITORIA

Los auditores deben tener la capacidad de validar los reportes y salidas y probar la autenticidad y precisión de los datos e información. Los de más interés son:

- Rastrear una transacción por cada paso del proceso y tener la capacidad de examinar los valores de datos intermedios producidos durante el procesamiento.
- imprimir registros y transacciones seleccionados del sistema que cumplan ciertos criterios (tales como una cuenta altamente activa o una cuenta con un saldo alto) para validar la precisión y autenticidad tanto de las transacciones como de los resultados.
- Mantener un balance constante en el sistema cuando éste implique cuestiones financieras y reportar sí el sistema está balanceado.
- Proporcionar los controles suficientes en la entrada, talen como controles y cuenta de los lotes y transacciones.

Auditoria debe conocer estos métodos, para que haya mayor conflabilidad de un sistema.

5.6.3.- CAPACIDAD

Se refiere al número de archivos que puede guardar y el número de archivos que puede conservar.

También se determina por.:

- El tamaño máximo de cada registro medido en bytes

- El tamaño máximo del archivo, medio en bytes
- El tamaño máximo del archivo, medio en campos por registro
- El número de archivos que pueden estar activos a la vez
- El número de archivos que se pueden registrar en un directorio de archivos.

5.6.4.- SOPORTE DEL PROVEEDOR

Se debe considerar los servicios que proporcionan. También necesita mantenimiento:

- ¿Con qué frecuencia se llevará a cabo el mantenimiento del software? ¿Se proporcionarán las nuevas versiones de manera regular? ¿Habrá un cargo por las actualizaciones?
- Si hay una cuota mensual por mantenimiento, ¿cuáles serán los servicios que cubrirá? ¿Cuáles serán los servicios que cubrirá? ¿Cuáles servicios se excluirán?
- El proveedor del software ¿Proporcionará la programación especializada para adecuar aspectos específicos del software a las necesidades demostradas del usuario? ¿A qué costo?
- ¿Con que prioridad? Es decir, qué tan pronto después de realizar el contrato comenzará y terminará la programación?
- ¿Cuáles son los acuerdos para controlar los aumentos en las cuotas de mantenimiento?

- ¿Cómo se resolverán los desacuerdos acerca de las necesidades de mantenimiento?

- ¿Durante qué horas estarán disponibles los servicios de soporte de emergencias en caso necesario después de las horas hábiles?

5.6.5.- Contratos de software

Se pueden delinear dos tipos de contratos de software:

1. Bosquejear los términos de alquiler de un programa de software. Cuando las organizaciones adquieren software, a menudo sólo reciben el derecho de usarlo mediante una cuota, ya sea una vez o por cierto tiempo. Una licencia permite a la organización el uso indefinido del software. Pero el comprador no lo posee y no puede vender el paquete, regalarlo o distribuir copias.

2. Bosquejar los términos de un proyecto de programación, en donde la organización contrata a un proveedor independiente para que produzca software. Los términos pueden establecer que el trabajo se hará mediante una cuota fija (por el costo del trabajo de la que lo desarrolle, más un porcentaje específico del costo que se añade al costo, dándole al que lo desarrolla una ganancia) o por un monto fijo y una duración específica en horas o en días calendario.

5.6.6. - compatibilidad del software existente

Es muy importante que partamos de esta aseveración de que el software que se adquiera tenga afinidad con el software ya existente ya que tendríamos que evaluar una conversión que podría ser desastrosa en cuanto al tiempo de empatar las metodológicas y filosofías del software ya existente y por lo tanto es muy importante que sean convergentes.

5.6.7. - crecimiento y explotación

No cabe duda que el crecimiento se dará de acuerdo a la expectativa de crecimiento y de como pueda compartir mi información y como le pueda sacar el mejor provecho a través de explosión de la información en las distintas áreas y departamentos de forma que deberá de contar con las características de transacción amigable del software seleccionado.

5.6.8. - mantenimiento y soporte

El servicio a brindar en cuento a soporte y mantenimiento deberá de ser simple sencillo y sin dificultades preponderantes ya que se persigue con esto bajo las expectativas anteriores que la problemática sea mínima si es que existe dificultad de manera que hasta el mismo personal de la empresa pueda darlo y no atarse con el proveedor.

5.6.9. - sencillez para la instalación del producto

La instalación del producto deberá ser bajo un standard definido por el proveedor garantizando que los módulos liberados sean de acuerdo y de acorde a las especificaciones anteriores de modulación e integral ya que modulo que se instale sea probado en condiciones reales y a satisfacción del cliente.

5.6.10. - material didáctico proporcionado por el proveedor

Definitivamente el proveedor proporcionara los manuales técnicos y de usuarios para el mantenimiento posterior y servicio ya que marcara la pauta para la no dependencia del proveedor a la hora de implementar el software existente.

5.6.11. - cantidad de proveedores que producen el software

Se supone que mínimo de acuerdo a la ley de adquisiciones de productos y servicios deberán de existen tres proveedores que brinden el servicio con el fin de tener mas opciones de selección del producto y que ofrezcan mas opciones de servicio y precio así como calidad en el servicio es muy importante que tengamos varios proveedores del software de forma como ya se menciono que no nos casemos con un solo proveedor.

5.6.12. - costeo preliminar

Él echo tal vez definitorio en la elección de un software es el costo podrá ser muy bonito y eficiente pero si cuenta mas de lo que se proyecto gastar es difícil de adquirir de forma que el costo es muy importante hacerlo incluso cuando se cita a los proveedores para la demostración del producto en ese momento lo podemos

costear para determinar cuanto se invertirá y si la empresa los pagara de acuerdo a lo presupuestado claro que aquí cabria la posibilidad de negociar y convencer a la empresa de los beneficios al adquirir dicho producto y justificar hasta cierto punto su costo.

5.6.13. - confrontación de proveedores

Se menciono que es importante reunir a los proveedores para obtener información clara y precisa del producto así como las ventajas y desventajas de producto también la reputación de servicio y trayectoria en el mercado como un respaldo de la inversión.

5.6.14. - elaboración de un documento oficial

Se desarrollan ciertas memorias o documentos donde se establece el procedimiento para establecer la compra que se acordó cuales son los derechos y obligaciones de los mismos y de que forma se compartirá las responsabilidades de buen uso y buen servicio.

5.6.15.- Iniciar tramites de adquisición

Ya una vez acordado los puntos del estira y afloja las pruebas, demostraciones, ventajas etc. Y de estar de acuerdo con el producto no resta que iniciar los tramites de adquisición ya que se tiene que considerar licencias tiempos de entrega derechos de autor para el bueno uso y aplicación es cuando solo resta esperar el producto.

5.6.16. - recepción e inspección del equipo

Se tiene que recibir el producto en lo convenido para efecto de su prueba y verificación con los manuales con anterioridad (técnico y usuario) para ponerlo en operación.

5.6.17. - registro del software

Se procede a registrar y asignar el software instalando a cada uno de sus usuarios involucrados en dicha adquisición para proceder a probarlo a tiempo real.

5.6.18. - resguardo de documentos

Se procede a guardar los documentos que serán el soporte de aclaraciones de lo establecido para que cumpla de acuerdo a la normatividad establecida.

5.6.19.- Supervisión de instalación

Es muy importante que personal de la empresa se encuentre en los momentos de la instalación para que en los momentos de algún conflicto sepan como resolverlo y no perder tiempos con la operatividad del mismo.

5.6.20. - capacitación para uso del equipo

El personal deberá de contar con la cultura suficiente y necesaria para la operatividad del producto y el buen uso del mismo ya que si no se producirán conflictos por el mal manejo se deberá de soportar con cursos de capacitación e instrucción.

5.6.21. - liberación del producto

Llego la hora de liberar un producto que se probo a conciencia y conocimiento de causa en lo que hace y debería de hacer solo requerimos que se libere el producto y para curarse en salud correr un paralelo con el fin de evitar problemas y tener un plan de contundencia listo.